DOCUMENT RESUME

ED 205 200 IR 009 504

AUTHOR	Kincaid, J. Peter: And Others
TITLE	Implementing the Computer Readability Editing System (CRES). Final Report.
INSTITUTION	Naval Training Analysis and Evaluation Group, Orlando, Fla.
REPORT NO	TAEG-R-98
PUB DATE	Mar 81
NOTE	178p.: For related documents, see ED 190 064 and IR 009 516.
EDRS PRICE	MF01/PC08 Plus Postage.
DESCRIPTORS	*Computer Oriented Programs: Computer Programs: *Editing: *Instructional Materials: *Material Development: Military Personnel: *Readability: *Technical Education

ABSTRACT

This report provides information on and guidelines for the implementation of the Computer Readability Editing System (CRES), which was developed for use in improving the ease of comprehending Navy technical manuals and training materials. The essential features of the CRES that potential users will have to consider in implementing the system in a new environment are described, as well as its use as a component of a computer-based publishing system. Step by step instructions accompanied by flow charts take the user through the five subsystems that support the two basic functions--the creation and maintenance of word lists and text files, and analysis of text passages. Appended documentation includes descriptions and programs for the master menu, text file subsystem, word substitution file subsystem, supplementary word file subsystem, phrase substitution subsystem, and text analysis programs. A list of the WANG Basic-2 commands frequently used in the programs provides explanations of their functions and examples. This report, together with TAEG-83 (ED 190 064), contains complete documentation of the system. (LLS)

****

* Reproductions supplied by EDRS are the best that can be made

* from the original document.

****

ED205200

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

TAEG Report No. 98

IMPLEMENTING THE COMPUTER READABILITY EDITING SYSTEM (CRES)

J. Peter Kincaid
Larry K. Cottrell
James A. Aagard
Peter Risley

Training Analysis and Evaluation Group

March 1981

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.

Alfred F. Smode
ALFRED F. SMODE, Ph.D., Director,
Training Analysis and Evaluation Group

W. L. Maloy
W. L. MALOY, Ed.D.
Deputy Chief of Naval Education and Training for Educational Development/
Research, Development, Test, and Evaluation

IR009504

ACKNOWLEDGMENTS

Several Training Analysis and Evaluation Group personnel participated in the writing of the computer program presented in this report, including Mr. Ray Cox, Mr. Charles Guitard, Ms. Carolyn Trotta, and Mrs. Nora Gregory.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER TAEG Report No. 98	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle) IMPLEMENTING THE COMPUTER READABILITY EDITING SYSTEM (CRES)	5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Peter Kincaid, Larry K. Cottrell, James A. Aagard, and Peter Risley	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Training Analysis and Evaluation Group Orlando, FL 32813	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE March 1981 13. NUMBER OF PAGES 177
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This report is intended to be a companion document to TAEG Report No. 83, Development and Test of A Computer Readability Editing System (CRES).

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Readability
Navy Vocabulary
Comprehension of Text
Computer Text Processing
Technical Manuals
Technical Writing
Training Manuals

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes how to implement the Computer Readability Editing System (CRES). As described in a previous TAEG report (No. 83), the CRES is a writer's aid to improve the comprehensibility of technical manuals and training materials. It contains features to (1) flag uncommon words and long sentences, (2) suggest replacements for awkward words and phrases, and (3) calculate the readability grade level. The CRES is now being implemented as part of computer-based publishing systems. (Continued on reverse)

20. ABSTRACT (continued)

This report contains the computer program flow charts described in that program, a user's guide, and a programmer's guide to help potential users.

TABLE OF CONTENTS

Section Page

I INTRODUCTION 3

Scope of Problem 3
Purpose of the Report 4
Organization of the Report 4

II IMPLEMENTING THE SYSTEM 5

Transportability of CRES 5

  Word Lists 5
  Hardware 5
  Software 7

Computer-Based Publishing Incorporating the CRES 7
CRES as an Element of the Naval Technical Information Presentation System 8

III GUIDE TO THE USE OF THE CRES 13

User's Guide 13

  Text File Subsystem 14
  Word Substitution Subsystem 17
  Supplementary Word Subsystem 21
  Phrase Substitution Subsystem 24
  Text Analysis Subsystem 27

REFERENCES 36

Appendix A Master Menu A-1
Appendix B Text File Subsystem B-1
Appendix C Word Substitution File Subsystem C-1
Appendix D Supplementary Word File Subsystem D-1
Appendix E Phrase Substitution File Subsystem E-1
Appendix F Text Analysis Programs F-1
Appendix G WANG BASIC-2 Commands G-1

LIST OF ILLUSTRATIONS

Figure		Page
1	Flow Chart Showing Phases of Editing by the Computer Readability Editing System.	6
2	First CRES Analysis to Add Words to Common Word List	9
3	Second CRES Analysis to Rewrite Original Text	10
4	Third CRES Analysis to Determine Readability Grade Level of Final Text	11
5	Major Components of the NTIP System	12
6	Flow Chart for Master Menu.	15
7	Flow Chart for Text Subsystem	16
8	Flow Chart for Word Substitution Subsystem.	19
9	Flow Chart for Supplementary Word Subsystem	23
10	Flow Chart for Phrase Substitution Subsystem.	26
11	Flow Chart for the Text Analysis Subsystem.	29
12	Flow Chart for Program XXANAL04	32
13	Flow Chart for the Phrase Analysis Subroutine of XXANAL04.	33
14	Flow Chart for the Word Analysis Subroutine of XXANAL04.	34
15	Flow Chart for the Print Subroutine of XXANAL04	35

SECTION I

INTRODUCTION

SCOPE OF PROBLEM

The Navy relies heavily on technical documents for both training and maintenance of equipment. According to figures tabulated by the Naval Technical Information Presentation Program (NTIPP)[1], the Navy annually produces 3 million pages of technical documents to add to an inventory approaching 40 million pages.

Unfortunately, Navy technical documents are often too difficult for enlisted personnel to easily use and understand. According to two recent studies conducted by the General Accounting Office (GAO):

- A growing number of military enlisted personnel have deficient reading skills (GAO, 1977).
- Technical documents for the U.S. military services are difficult to read and use and it will take a major and costly effort to rewrite them to enable the recruits of the 1980s to understand them (GAO, 1979).

The Chief of Naval Education and Training (CNET), in recognition of these problems, tasked the Training Analysis and Evaluation Group (TAEG) to undertake the development of the Computer Readability Editing System (CRES).[2]

TAEG Report No. 83[3] describes the development and test of the CRES and provides detailed examples of its use. The system is designed to aid authors in writing comprehensible text and contains a number of specific features, each designed to achieve this goal.

- Long sentences are flagged and the number of words in the sentence is shown.
- Replacements for awkward words and phrases are suggested.
- Uncommon words are flagged.
- Keying errors and misspelled words are flagged.
- The readability grade level calculated according to the Department of Defense readability standard, the Flesch-Kincaid Formula, is shown.

---

[1] Personal communication, S. C. Rainey, Technical Manager, NTIPP.

[2] CNET ltr of 29 June 1978.

[3] J. Peter Kincaid, James A. Aagard, and John W. O'Hara. Development and Test of a Computer Readability Editing System (CRES). TAEG Report No. 83, March 1980. Training Analysis and Evaluation Group, Orlando, FL 32813

Numerous agencies producing technical documents for the Navy and other military services have requested detailed information on how to implement the CRES at their own facilities. In a related development, NTIPP, charged with the development of a high-technology publishing system for the Navy during the mid-1980s and beyond, plans to incorporate key features of the CRES in its overall system. Consequently, documentation of hardware, software, and operational requirements for CRES was considered necessary to assist potential users of the system.

PURPOSE OF THE REPORT

The purpose of this report is to provide information and guidelines on the implementation of the CRES. Together with TAEG Report No. 83 it contains complete documentation of the system, including the computer program.

ORGANIZATION OF THE REPORT

In addition to this introduction, the report contains two sections and seven appendices. Section II describes ways to implement the CRES. It contains a description of the system as part of computer-based publishing systems. Section III contains flow charts and associated descriptions of the computer program. The reader is referred to TAEG Report No. 83 for the various word lists contained in the CRES and a detailed description of the operation of the system. Appendix A contains the program listing for START, the master menu of the system. Appendix B contains the program listing for the Text File Subsystem. Appendix C contains the program listing for the Word Substitution File Subsystem. Appendix D contains the program listing for the Supplementary Word File Subsystem. Appendix E contains the program listing for the Phrase Substitution File Subsystem. Appendix F contains the program listing for the Text Analysis programs. Appendix G contains a glossary of WANG BASIC-2 commands.

SECTION II

IMPLEMENTING THE SYSTEM

This section provides a description of the essential features of the CRES that potential users will have to consider in implementing the system in a new environment. Also described is the use of the CRES as a component of computer-based publishing systems.

Figure 1 provides an overview of the CRES. This figure identifies key components of the system and serves as a reference for the discussion which follows.

TRANSPORTABILITY OF CRES

The CRES is presently a prototype system which operates on a WANG 2200 MVP computer. While it has been extensively used for demonstrations and a limited amount of production, it is yet to be used for a large-scale production effort. Facilities which adopt the system will need to consider changing the word lists, hardware, and software to fit their own needs.

WORD LISTS. The word lists in the CRES consist of common words, technical words, and a word and phrase substitution dictionary which is general in nature. They have been prepared and tested to apply to a wide variety of military technical training and technical manual texts. Specialized text (e.g., a technical manual for a specific piece of electronics equipment) will almost certainly include, in addition to the words in the existing lists, additional specialized terms. Most publishing centers that might use the CRES will publish materials that incorporate specialized vocabularies which will have to be added to the system. For example, the U.S. Army limits vocabulary in many training and technical manuals to an Army common word list (CIR 310-9); the CRES could easily use this list.

HARDWARE. The existing version of the CRES uses WANG computer equipment. However, many makes of computer equipment could be used provided the following is included: a central processing unit (CPU) with 64 kilobytes of memory, a disk platter with a minimum of 5 megabytes of memory, a CRT, and a printer. The specific hardware used in the existing system includes:

- WANG 2236D CRT for entry and display
- WANG 2200 MVP CPU
- WANG 2200 compatible printer
- 13.4 megabytes disk platter and compatible disk drive.

Supplementary hardware available for maintenance and transportability of the CRES includes an 8" floppy diskette drive and an 800 bpi 9 track tape drive.

The system was designed to operate using a 64 kilobyte central memory as this is a common size. However, this memory limitation required special programming techniques in developing the program.

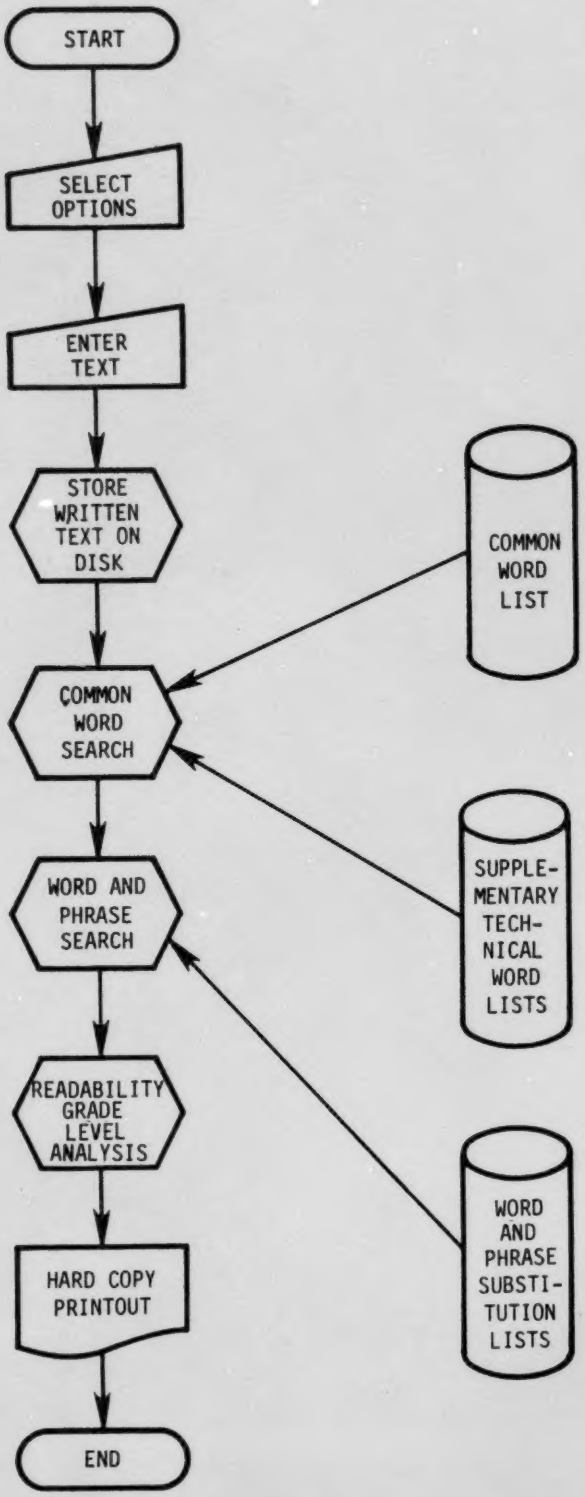
START
SELECT OPTIONS
ENTER TEXT
STORE WRITTEN TEXT ON DISK
COMMON WORD LIST
COMMON WORD SEARCH
SUPPLEMENTARY TECHNICAL WORD LISTS
WORD AND PHRASE SEARCH
READABILITY GRADE LEVEL ANALYSIS
WORD AND PHRASE SUBSTITUTION LISTS
HARD COPY PRINTOUT
END

Figure 1. Flow Chart Showing Phases of Editing by the Computer Readability Editing System

All software (programs) and data (word lists) reside on a 13.4 megabyte platter of the disk drive. Fifty-six kilobytes of central memory of the CPU are allocated for operation of the system. The 2236D CRT can display a maximum of 24 rows by 80 columns of characters for a total of 1,920 characters.

Excluding the tape drive, the current cost of the hardware is about $25,000.

SOFTWARE. The computer program for the CRES is written in WANG BASIC-2 and has not yet been translated into any other programming language. Publishing centers with other types of computer equipment who adopt the CRES can expect to spend time (perhaps two man-months) rewriting the program. The flow charts contained in figures 6 through 15 and the description of the program subsystems contained in appendices A through G will aid considerably in this task.

COMPUTER-BASED PUBLISHING INCORPORATING THE CRES

Many companies and agencies that produce technical documents for the U.S. military services have converted to computer-based publishing systems (Keeler, 1977). Such systems use computer word processing equipment for recording, revising, and formatting text and for typesetting camera-ready copy. Typically, computer processing of the text is preceded by typing at least one draft using a conventional typewriter.

Given an ever-increasing reliance on computers for publishing, the use of computerized authoring and editing aids requires no large additional investment in computer equipment. In addition, adding the CRES to a computer-based publishing system will probably save time for authors and editors. The system suggests easily made revisions to the text and flags most misspelled words. Compared with the procedures used with existing word processing systems, the text is simply entered at an earlier stage and editing is done in the CRES using the same textstream. The edited textstream is used to output camera-ready copy.

The major advantage of the system is as an aid in producing comprehensible text for enlisted personnel with limited reading skills. The requirement that text be written at specified readability grade levels is now being imposed on contractors. For example, training materials currently being written for the Royal Saudi Naval Forces are required to be no higher than the seventh grade readability level.

The TAEG is currently producing an eighth grade level text using a computer-based system including the CRES. The text is a workbook to be used by high school students enrolled in a third year Naval science course. The initial draft of the material is being analyzed and revised using the CRES. Then it will be printed in camera-ready form from the same textstream using a high quality daisy wheel printer.

The use of the CRES with text employing specialized vocabulary is illustrated in figures 2, 3 and 4. These figures show successive analyses of the same text which is taken from technical manuals for the TRIDENT Submarine.

Figure 2 (the first analysis) shows the flagging of several technical terms that should be known by specialists who would use this particular technical manual. The specialized terms to be added to the common word list are indicated by a check mark, for example, "bellows-piston" and "dissipator." The technical manual should include illustrations of these and other specialized parts. Other specialized terms could be defined in a glossary to be included in the technical manual and used as a training aid.

Figure 3 (the second analysis) was completed after the specialized terms checked in figure 2 were added to the common word list. These terms are consequently not flagged as uncommon. Editing, based on computer-generated suggestions, is also shown. The passage consists of two long sentences of 32 and 31 words which are logically broken into shorter sentences. The substitution of "sending" and "found" for the original words "transmitting" and "located" is also shown. These easier substitutes are automatically printed by the CRES.

Figure 4 (the third and final analysis) is necessary to determine the readability grade level of the text to be included in the technical manual. Note that the readability grade level has been reduced from 17.4 (for the original text) to 11.0 (for the rewritten text).

Nearly every new kind of text will require some modification in word lists. Since the CRES printout lists "uncommon" words (those not on the common word list) at the end of each analysis, this is an easy process. An editor would probably make the final decision as to which words to add to the word lists once a sufficient amount of text has been analyzed (perhaps 5,000 to 10,000 words). The program is written so that words can easily be added to the various lists.

CRES AS AN ELEMENT OF THE NAVAL TECHNICAL INFORMATION PRESENTATION SYSTEM[3]

The Naval Technical Information Presentation System (NTIPS) of the David W. Taylor Naval Ship Research and Development Center has as its goal the improvement of the Navy's efficiency in publishing technical information to support Navy equipment. The program is concerned with producing technical information needed for training, maintenance, equipment operation, and logistic support. The program aims are: (1) to define a full set of hardware support documents including job performance aids and learning aids and (2) to design a high-technology system for authoring, editing, composing, illustrating, printing, distributing, and updating these documents. The major subsystems of the NTIPS are shown in figure 5.

---

[3] Much of this section is based on a presentation by Hughes Aircraft Company to the NTIPS Technical Manager, June 1980.

```
  The temperature [sensor] for the heat [dissipator] is a [gas-filled] bulb
which responds to temperature changes by [transmitting][*SENDING*] a
[proportional] change in its internal pressure to a [bellows] [located][*FOUND]
within the valve.$$32$$ The [bellows] is mechanically linked to the water valve
piston so that, in responding to the change in pressure, the [bellows-piston]
assembly effectively regulates the flow of [coolant] through the
[dissipator.]$$31$$

------------------      READABILITY RESULTS      ------------------------
Number of Sentences         Number of Words           Number of Syllables
        2                          63                          111

Avg. Number of Words per Sentence      Avg. Number of Syllables per Word
                31.50                                  1.76

GRADE LEVEL      (Based on DOD Readability Standard)
     17.4

------------------ WORDS NOT ON COMMON WORD LISTS ------------------------
WORD                    FREQ           WORD                      FREQ

✓bellows                  2            ✓bellows-piston             1
✓coolant                  1            ✓dissipator                 2
✓gas-filled               1             proportional               1
✓sensor                   1
```

Add words with ✓ to common word list.

Figure 2. First CRES Analysis to Add Words to Common Word List

The temperature sensor for the heat dissipator is a gas-filled bulb. It ~~which~~ responds to temperature changes by ~~[transmitting]~~ [*SENDING*] a [proportional] change in its internal pressure to a bellows ~~[located]~~ [*FOUND*] within the valve.$$32$$ The bellows is mechanically linked to the water valve piston; ~~so that~~ in responding to the change in pressure, then bellows-piston assembly effectively regulates the flow of coolant through the dissipator.$$31$$

```
-------------------      READABILITY RESULTS      ------------------------

Number of Sentences        Number of Words          Number of Syllables
        2                        63                         111

Avg. Number of Words per Sentence      Avg. Number of Syllables per Word
               31.50                                   1.76

GRADE LEVEL     (Based on DOD Readability Standard)
     17.4

----------------- WORDS NOT ON COMMON WORD LISTS ----------------------

WORD                   FREQ              WORD                     FREQ

proportional             1
```

Figure 3. Second CRES Analysis to Rewrite Original Text

```
  The temperature sensor for the heat dissipator is a gas-filled bulb.  It
responds to temperature changes by sending a [proportional] change in its
internal pressure to a bellows found within the valve.  The bellows is
mechanically linked to the water valve piston: in responding to the change in
pressure, the bellows-piston assembly effectively regulates the flow of coolant
through the dissipator.

------------------      READABILITY RESULTS      ----------------------
Number of Sentences        Number of Words          Number of Syllables
         4                        61                        107

Avg. Number of Words per Sentence     Avg. Number of Syllables per Word
               15.25                                  1.75

GRADE LEVEL     (Based on DOD Readability Standard)
      11.0
------------------WORDS NOT ON COMMON WORD LISTS-----------------------
WORD                FREQ              WORD                      FREQ
proportional         1
```

Figure 4. Third CRES Analysis to Determine Readability Grade Level of Final Text

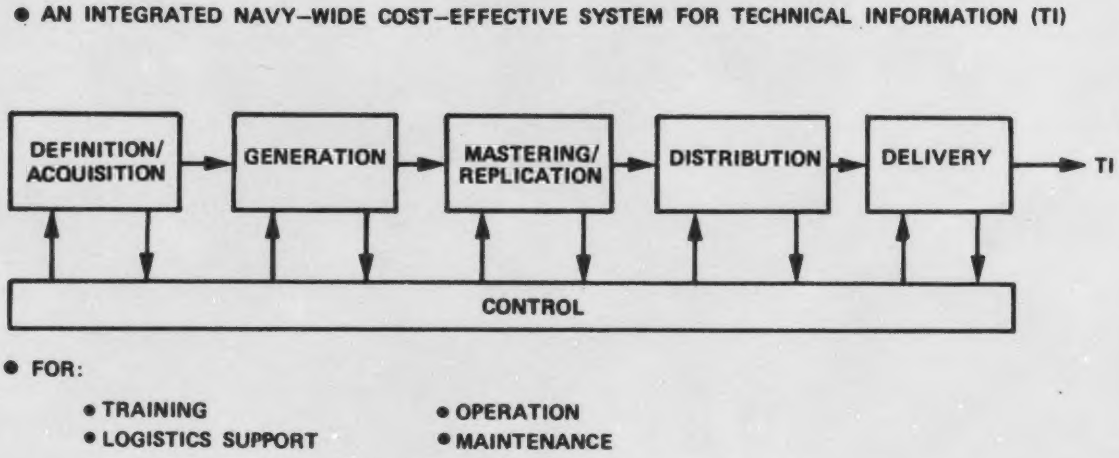
AN INTEGRATED NAVY-WIDE COST-EFFECTIVE SYSTEM FOR TECHNICAL INFORMATION (TI)
DEFINITION/ ACQUISITION
GENERATION
MASTERING/ REPLICATION
DISTRIBUTION
DELIVERY
TI
CONTROL
FOR:
TRAINING
LOGISTICS SUPPORT
OPERATION
MAINTENANCE

Figure 5. Major Components of the NTIP System

The CRES is simply one component of the authoring subsystem which also includes such things as the generation of graphics, effective integration of text and graphics, and keeping track of part numbers and nomenclature. Specifically, the purpose of the NTIPS authoring subsystem is to provide computer assistance to authors writing and editing technical information through a system which allows:

- enforcement of technical information specifications (e.g., content, format, and style specifications)
- use of writers who do not need to have detailed knowledge of formatting requirements.

Automation should result in standardization, reduction of errors introduced by repeated manual actions, and consequent cost savings.

The NTIPS computer-assisted authoring system must have the capability to:

- handle classified data
- allow sharing of tasks and information by multiple authors
- allow on-line editing of text and graphics
- provide on-line author training and assistance
- support various sized contractor authoring facilities, including those who can afford to make a capital investment in an authoring system as well as those who cannot
- output digitized technical information for subsequent use in mastering and replication.

SECTION III

GUIDE TO THE USE OF THE CRES

This section contains step-by-step directions for using the CRES. The system is designed to be highly interactive, and instructions are provided throughout to aid the user. A potential user of these procedures should make sure that the CRES is stored on a specific disk of the hardware system being used. The task of transferring the CRES from tape to disk is usually accomplished by a programmer. The user of the CRES should consult with the programmer in order to determine the disk on which the CRES resides.

The CRES supports two basic functions. These are: (1) the creation and maintenance of word lists and text files and (2) analysis of the text passages.

The software required for creating and maintaining word lists and text files is divided into five subsystems. These include:

- Text File Subsystem
- Word Substitution Subsystem
- Supplementary Word Subsystem
- Phrase Substitution Subsystem
- Text Analysis Subsystem.

USER'S GUIDE

Access to a specific subsystem is accomplished by the following procedure. First press the "RESET" key. Then press the "CLEAR" key followed by C/R (C/R stands for carriage return). ALL ENTRIES SHOULD BE FOLLOWED BY PRESSING C/R. Next select the device address of the programs by keying in "SELECT DISK XXX", followed by C/R. "XXX" stands for the device address where the programs are stored. A programmer will give you the correct disk addresses for your computer. Next key in "LOAD RUN", C/R. The following menu will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
             * * * READABILITY SYSTEM MASTER MENU * * *

OPTIONS
   1      -    LOAD TEXT FILE SUBSYSTEM
   2      -    LOAD WORD SUBSTITUTION SUBSYSTEM
   3      -    LOAD SUPPLEMENTARY WORD SUBSYSTEM
   4      -    LOAD PHRASE SUBSTITUTION SUBSYSTEM
   5      -    LOAD TEXT ANALYSIS SUBSYSTEM

  99      -    ** End of Session **

ENTER OPTION?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The computer program for producing the above master menu is START. The flow chart which depicts the flow of control in program START is given in figure 6. Program START is contained in appendix A.

TEXT FILE SUBSYSTEM. The text file subsystem allows the user to input, edit or print a passage or a file of text. A passage of text is defined as one screen full of text or 25 lines. A text file is one or more passages that go together.

To start the text file subsystem enter "1" as the option number for the Master Menu. After entering "1" the text file subsystem menu will be displayed as follows:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                * * * TEXT FILE SUBSYSTEM MENU * * *

OPTIONS

   1      -    LOAD TEXT INPUT PROGRAM
   2      -    LOAD TEXT EDIT PROGRAM
   3      -    LOAD TEXT PRINT PROGRAM

  99      -    ** Return to Master Menu **

ENTER OPTION?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 7 displays the flow of control for the submenu START1. The computer programs which support the text file subsystem are given in appendix B.

Using The Text File Subsystem. To input a passage enter "1" as the option number followed by C/R. The following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE FOR THE TEXT STORAGE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Now enter the device address of the device where you wish to store the text passage. For example, "D33".

Next the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER YOUR TEXT NOW (TOUCH STMNT NBR/FN KEY TO EXIT)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You can now enter the text. The text can be edited by moving the cursor up and down, or back and forth. You can also insert or delete characters by pressing the Special Function keys. When finished, touch the Statement No. key (Wang VP system) or FN key (Wang MVP system). You can fill all or part of the screen with text, but you can not enter more than one screen of text before pressing the Statement No. or FN key.

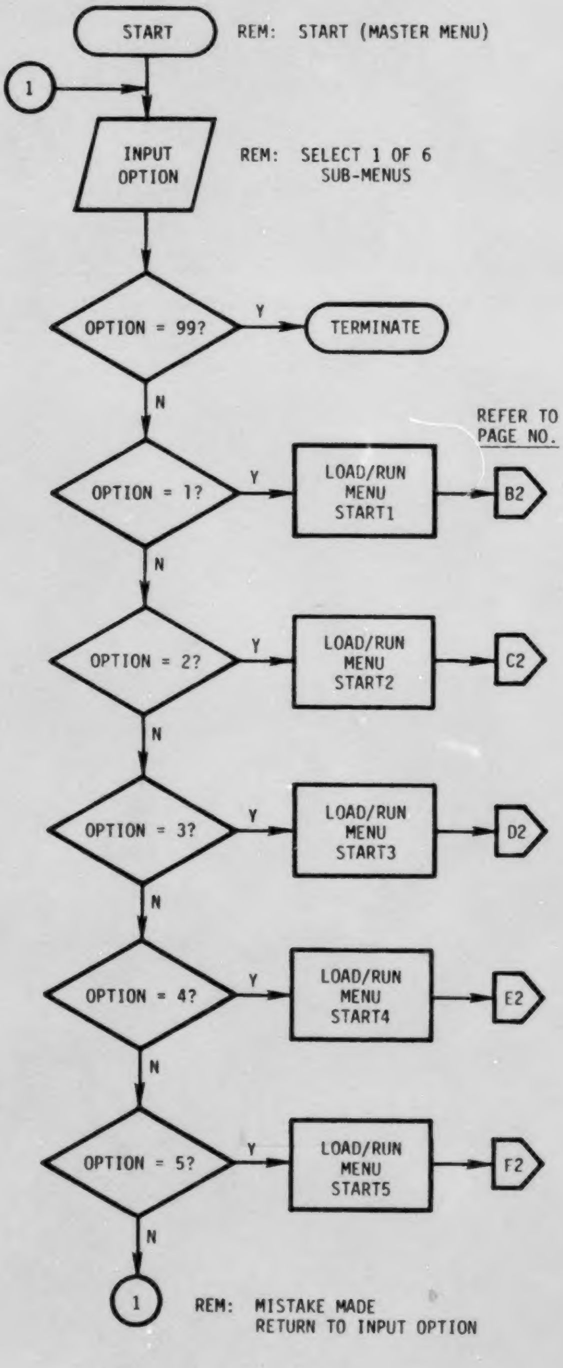
START
REM: START (MASTER MENU)
1
INPUT OPTION
REM: SELECT 1 OF 6 SUB-MENUS
OPTION = 99?
Y
TERMINATE
N
REFER TO PAGE NO.
OPTION = 1?
Y
LOAD/RUN MENU START1
B2
N
OPTION = 2?
Y
LOAD/RUN MENU START2
C2
N
OPTION = 3?
Y
LOAD/RUN MENU START3
D2
N
OPTION = 4?
Y
LOAD/RUN MENU START4
E2
N
OPTION = 5?
Y
LOAD/RUN MENU START5
F2
N
1
REM: MISTAKE MADE RETURN TO INPUT OPTION

Figure 6. Flow Chart For Master Menu

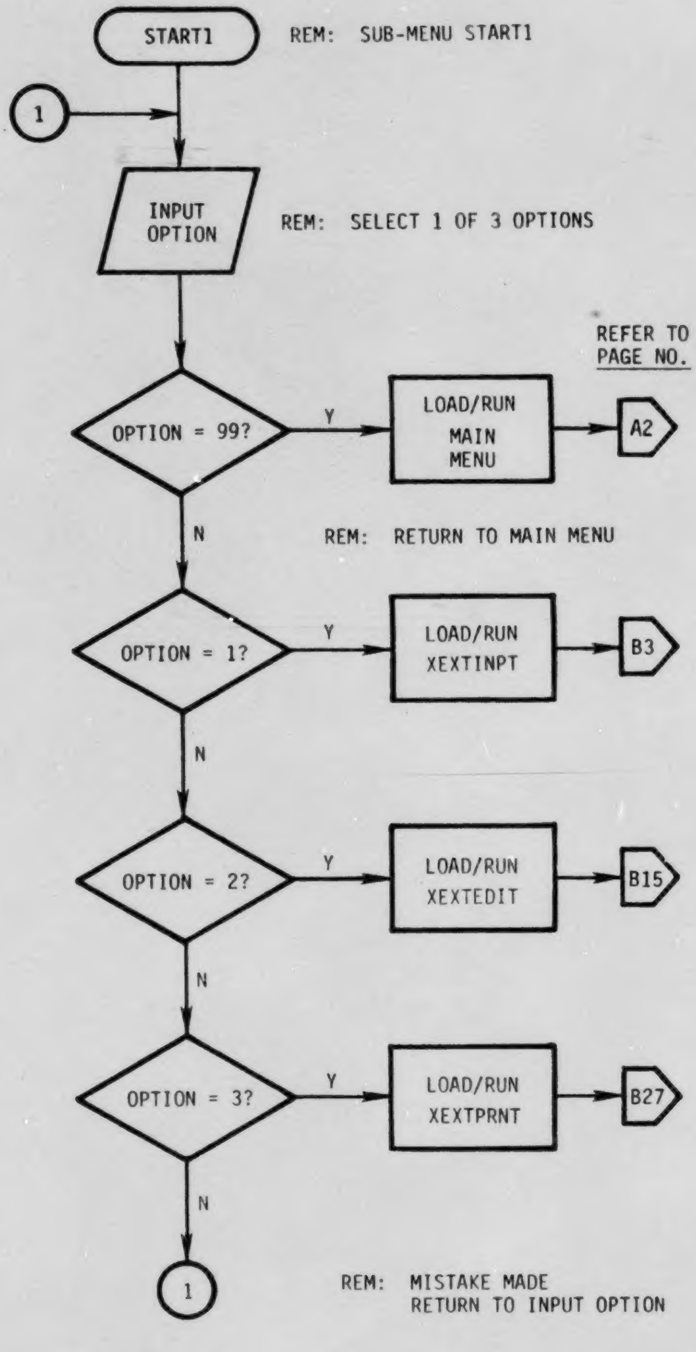
START1
REM: SUB-MENU START1
1
INPUT OPTION
REM: SELECT 1 OF 3 OPTIONS
REFER TO PAGE NO.
OPTION = 99?
Y
LOAD/RUN MAIN MENU
A2
N
REM: RETURN TO MAIN MENU
OPTION = 1?
Y
LOAD/RUN XEXTINPT
B3
N
OPTION = 2?
Y
LOAD/RUN XEXTEDIT
B15
N
OPTION = 3?
Y
LOAD/RUN XEXTPRNT
B27
N
1
REM: MISTAKE MADE
RETURN TO INPUT OPTION

Figure 7. Flow Chart For Text Subsystem

An up arrow "↑" in column 1 of a line of text will cause the analysis program to print that line exactly the way it is typed without analyzing it. This feature is useful for printing tables or other text that you do not want to be analyzed. If you wish to have one or more chapter headings then put the heading in angled brackets. For example,<Section One>. This will cause the analysis program to skip to a new page and print the heading at the top of the page.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      DO YOU WISH TO SAVE THIS TEXT ON DISK?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If the answer is "N", then the subsystem menu will be loaded again and the current text is permanently lost. If the answer is "Y", then the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      ENTER THE FILE NAME OF THE TEXT FILE TO SAVE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The file name should be eight characters long and each passage should have a different name. The file name should be in the format "SSSSnnnn"; where "SSSS" is the file identifier and "nnnn" is the sequence number of the passage within that file. The file identifier must be the same for all passages in the same file. For example: If you have three continuous passages in one file they could be called ABCD0100, ABCD0200, and ABCD0300. If you want to insert a passage between two passages that are already on a file then you should give the new passage a sequence number that falls between the sequence numbers of the existing passages. For example: If you created a new passage with the file name ABCD0150 then it would automatically be inserted between ABCD0100 and ABCD0200. You must not give a new passage of text the sequence number 0000. The sequence number 0000 is reserved to identify a group of passages that make up a text file. For example: If you wish to print or analyze all the passages with file identifier ABCD then you would enter the name ABCD0000 when the text print or analysis programs ask for the file name.

Edit. To edit a passage that already exists on file, enter "2" as the answer to the "ENTER OPTION" question in the subsystem menu. Then the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      ENTER THE DEVICE FOR TEXT
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

WORD SUBSTITUTION SUBSYSTEM. To call up the word substitution program, key "2" as the option number in the Master Menu. The following subsystem menu will be displayed:

```
-----------------------------------------------------------------
          * * *  WORD SUBSTITUTION SUBSYSTEM MENU * * *

     OPTIONS

         1      -    LOAD WORD SUBSTITUTION FILE INPUT PROGRAM
         2      -    LOAD WORD SUBSTITUTION FILE EDIT PROGRAM
         3      -    LOAD WORD SUBSTITUTION FILE PRINT PROGRAM
         4      -    LOAD ADD EXPANDED ROOT WORDS PROGRAM

        99      -    ** Return to Master Menu **

     ENTER OPTION
-----------------------------------------------------------------
```

Figure 8 gives the logic for the submenu START2. The computer programs which support the word substitution subsystem are contained in appendix C.

Using The Word Substitution Subsystem. To input a new word substitution file, enter "1" as the option number in the subsystem menu. The following message will be displayed:

```
-----------------------------------------------------------------
     ENTER THE DEVICE OF THE WORD SUBSTITUTION FILE
-----------------------------------------------------------------
```

Now enter the device address where you want to store the word substitution file. Next, the following message will be displayed:

```
-----------------------------------------------------------------
     ENTER THE DEVICE OF THE WORK FILE
-----------------------------------------------------------------
```

Enter the device address for the temporary work file. (If you are unsure which device to use, ask a programmer.) Next, the following message will be displayed:

```
---------------------------------------------------------------
     ENTER THE FILE NAME OF THE BLOCK POINTER FILE
---------------------------------------------------------------
```

Enter the file name for the block pointer file. This is a unique eight character name. Next, the following message will be displayed:

```
---------------------------------------------------------------
     ENTER THE FILE NAME OF THE WORD SUBSTITUTION FILE
---------------------------------------------------------------
```

Enter the file name you wish to use for the word substitution file. The block pointer file name just entered should always be used with the word substitution file name you just entered. Next, the following message will be displayed:

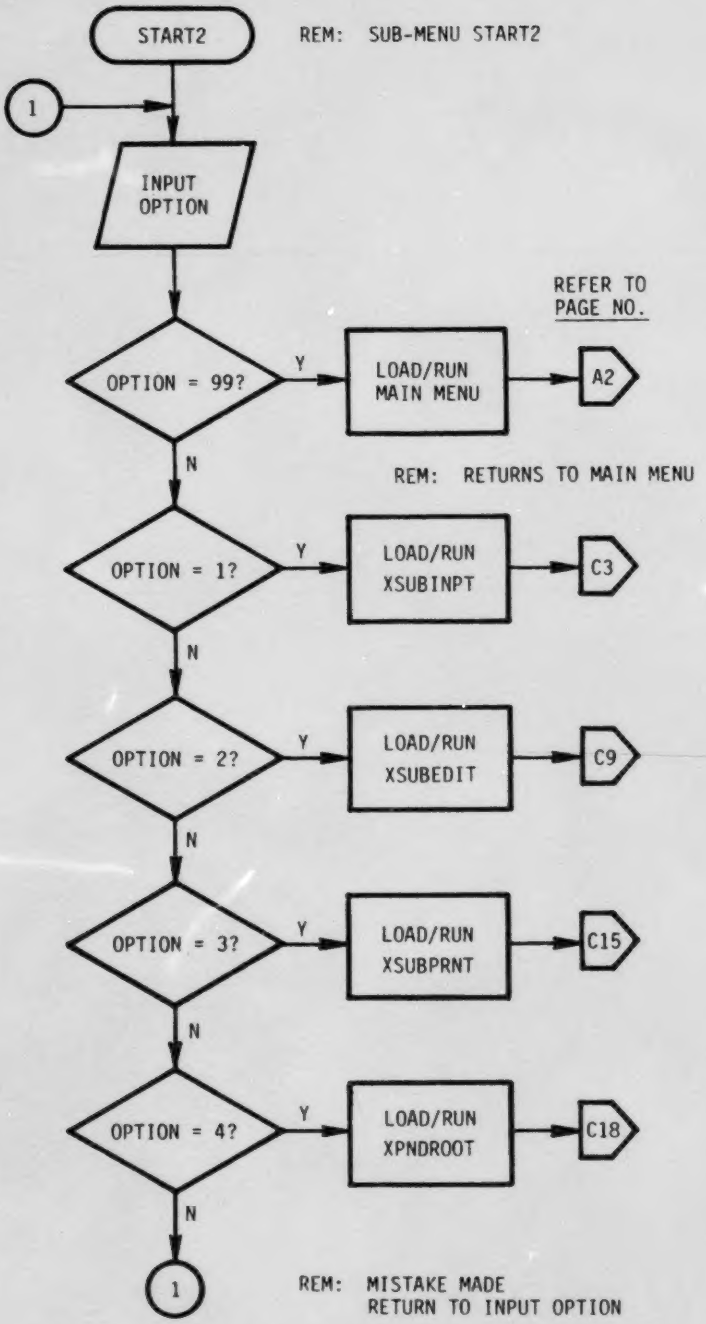
START2
REM: SUB-MENU START2
1
INPUT OPTION
REFER TO PAGE NO.
OPTION = 99?
Y
LOAD/RUN MAIN MENU
A2
N
REM: RETURNS TO MAIN MENU
OPTION = 1?
Y
LOAD/RUN XSUBINPT
C3
N
OPTION = 2?
Y
LOAD/RUN XSUBEDIT
C9
N
OPTION = 3?
Y
LOAD/RUN XSUBPRNT
C15
N
OPTION = 4?
Y
LOAD/RUN XPNDROOT
C18
N
1
REM: MISTAKE MADE RETURN TO INPUT OPTION

Figure 8. Flow Chart For Word Substitution File Subsystem Menu

```
     ENTER WORD OR EOL
```

Now enter the word you wish to add to the word substitution file. If the word has already been entered then the following message will be displayed:

```
     WORD IS ALREADY ON FILE
     ENTER 1) TO DELETE WORD  2) TO CHANGE SUBSTITUTES  3) TO
     LEAVE WORD UNCHANGED
```

Enter "1" to delete the word you just entered or enter "2" if you wish to change the substitutes or enter "3" to leave the word unchanged. If the word is not already on the file then the following message will be displayed:

```
     ENTER FIRST SUBSTITUTE OR C/R
```

Enter the first substitute for the word just entered or press C/R if the word does not have a substitute. Next, the following message will be displayed:

```
     ENTER SECOND SUBSTITUTE OR C/R
```

Enter the second substitute for the word just entered or press C/R if the word does not have a substitute.

Repeat the above process until you have entered all the words you want to enter. Then enter "EOL" to return to the subsystem menu.

Edit. To edit a word substitution file that already exists, enter "2" as the option number of the subsystem menu. The procedure for editing a word substitution file is exactly the same as the procedure to input a word substitution file except that the word substitution file and the block pointer file must already exist for this program. You can add, delete, or change words on the existing file with this program.

Print. To print a word substitution file enter "3" as the option number of the subsystem menu. The following message will then be displayed:

```
     ENTER DEVICE FOR FILE STORAGE
```

Enter the device address where the word substitution file is stored. Next, the following message will be displayed:

```
     ENTER FILE NAME OF WORD SUBSTITUTION FILE
```

Enter the file name of the word substitution file you wish to print. Next, the following message will be displayed:

```
ENTER THE DEVICE FOR PRINTING LIST
```

Enter the device address of the printer you wish to have the file printed on. Next, the following message will be displayed:

```
DO YOU WISH TO TITLE THIS LIST? (Y/N)
```

Enter "Y" for yes or "N" for no. If you answer yes then you can enter a title of up to 40 characters. The file will now begin printing and when it is done the subsystem menu will be displayed.

Add Root and Expanded Word. Enter "4" as the option number in the subsystem menu to add root words and their expanded forms to a word substitution file (which is made up of both the common word list and the word substitution list). A root word is a basic word without any endings like s, es, ed, ing, etc. You will then be asked to enter the device address of the word substitution and work files. You will also be asked the file name of the word substitution file and the block pointer file. Next, you will be asked to enter the root word or "EOL". When you are finished, key in "EOL" and press C/R. Otherwise, key in the root word. The root word followed by all of its possible endings will be displayed. Some of these words will be nonsense words. For example: If you enter the word "stand", some of the words on the expanded list would be "standed", "standily", and "standy". When the list is displayed, the first word on the list will be underlined. You can modify or change a word while it is underlined. If you want to have an underlined word added to the word substitution file the way it is shown typed then just press C/R. The next word in the list will then be underlined. If you want to delete the underlined word from the list press ERASE and then C/R. You can also change an underlined word by writing over top of it and then pressing C/R. For example, "standed" could be changed to "stood". When you have gone through the whole list the program will ask if the list is correct or not. If you wish to go through the list again then enter "N". If you enter "Y" then the list that is on the screen will be added to the word substitution file and you will be asked to enter the next root word.

Note: You can not enter any suggested substitutes with this program. If you wish to add substitutes for a word then you must use the word substitution file edit program.

SUPPLEMENTARY WORD SUBSYSTEM. The supplementary word subsystem will input, edit or print a supplementary word file. Select option number "3" of the Master Menu to use this subsystem. The following menu will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            * * * SUPPLEMENTARY WORD FILE SUBSYSTEM * * *

     OPTIONS

        1      -    LOAD SUPPLEMENTARY WORD FILE INPUT PROGRAM
        2      -    LOAD SUPPLEMENTARY WORD FILE EDIT PROGRAM
        3      -    LOAD SUPPLEMENTARY WORD FILE PRINT PROGRAM

       99      -    ** Return to Master Menu **

     ENTER OPTION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The flow of control for the submenu START3 is displayed in figure 9. Appendix D contains the computer programs which constitute the supplementary word subsystem.

To input a new supplementary word list enter "1" as the option number of the subsystem menu. Then the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE OF THE SUPPLEMENTARY WORD FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the device address where you wish to store the new supplementary word file. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE OF THE WORK FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the device address of the temporary work file. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE FILE NAME OF THE SUPPLEMENTARY WORD FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the file name of the supplementary word file you wish to create. It must be a unique name of up to eight characters. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER WORD OR EOL
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the word you want to add to the supplementary word list or enter "EOL" if you are done. If the word has already been entered then the following message will be displayed:

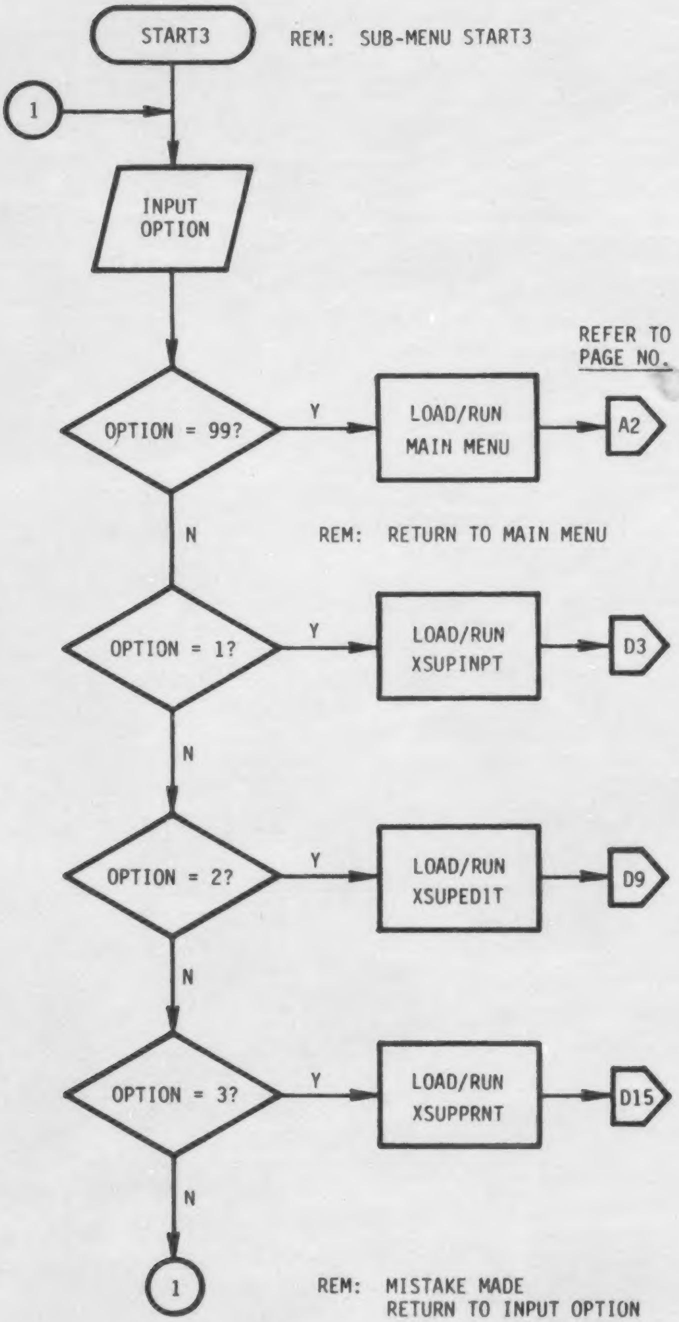
START3
REM: SUB-MENU START3
1
INPUT OPTION
REFER TO PAGE NO.
OPTION = 99?
Y
LOAD/RUN MAIN MENU
A2
N
REM: RETURN TO MAIN MENU
OPTION = 1?
Y
LOAD/RUN XSUPINPT
D3
N
OPTION = 2?
Y
LOAD/RUN XSUPEDIT
D9
N
OPTION = 3?
Y
LOAD/RUN XSUPPRNT
D15
N
1
REM: MISTAKE MADE
RETURN TO INPUT OPTION

Figure 9. Flow Chart For Supplementary Word Subsystem

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     WORD IS ALREADY ON FILE
     ENTER 1) TO DELETE WORD 2) TO LEAVE WORD UNCHANGED
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter "1" to take the word out of the file or "2" to leave the word in the file. If you want to change a word then you must delete the old word and enter the word the way you would like to have it.

Edit. To edit a supplementary word file that already exists, enter "2" as the option number for the subsystem menu. The edit program works exactly the same way as the input program except that the supplementary word file you want to edit must already exist. You can add or delete words with this program.

Print. To print a supplementary word file enter "3" as the option number for the subsystem menu. The following message will then be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER DEVICE FOR FILE STORAGE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the device address where the supplementary word file is stored. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER FILE NAME OF SUPPLEMENTARY WORD FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the file name of the supplementary word file you wish to print. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER DEVICE FOR PRINTING LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the device address of the printer you wish to use to print out the supplementary word file. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     DO YOU WISH TO TITLE THIS LIST? (Y/N)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter "Y" for yes or "N" for no. If you answer yes then enter a title of up to 40 characters. The supplementary word file will now print out and when it is finished the subsystem menu will be displayed.

PHRASE SUBSTITUTION SUBSYSTEM. The phrase substitution subsystem allows the user to input, edit, or print a phrase substitution file. To start the phrase substitution subsystem, enter "4" as the option number for the Master Menu. The following menu will then be displayed:

```
-------------------------------------------------
      * * * PHRASE SUBSTITUTION SUBSYSTEM MENU * * *

      OPTIONS

        1     -   LOAD PHRASE SUBSTITUTION FILE INPUT PROGRAM
        2     -   LOAD PHRASE SUBSTITUTION FILE EDIT PROGRAM
        3     -   LOAD PHRASE SUBSTITUTION FILE PRINT PROGRAM

       99     -   ** Return to Master Menu **

      ENTER OPTION
-------------------------------------------------
```

To input, edit, or print a phrase substitution file enter either "1", "2", or "3". Each phrase can be up to 30 characters long and each substitute can be up to 25 characters long. The maximum number of phrases that can be put on one file is 126.

Figure 10 gives the logic of the submenu START4. Consult appendix E for the computer programs which constitute the phrase substitution subsystem.

If you wish to input a new phrase substitution file enter "1" as the option number from the subsystem menu. The following message will then be displayed:

```
-------------------------------------------------
      ENTER THE DEVICE OF THE PHRASE SUBSTITUTION FILE
-------------------------------------------------
```

Enter the device address where you wish to store the new phrase substitution file. Next, the following message will be displayed:

```
-------------------------------------------------
      ENTER THE FILE NAME OF THE PHRASE SUBSTITUTION FILE
-------------------------------------------------
```

Enter the file name of the new phrase substitution file. The file name must be a unique name of up to eight characters. Next, the following message will be displayed:

```
-------------------------------------------------
      ENTER PHRASE OR EOL
-------------------------------------------------
```

Now enter the phrase you wish to add or "EOL" if you are finished. If the phrase just entered has already been entered then the following message will be displayed:

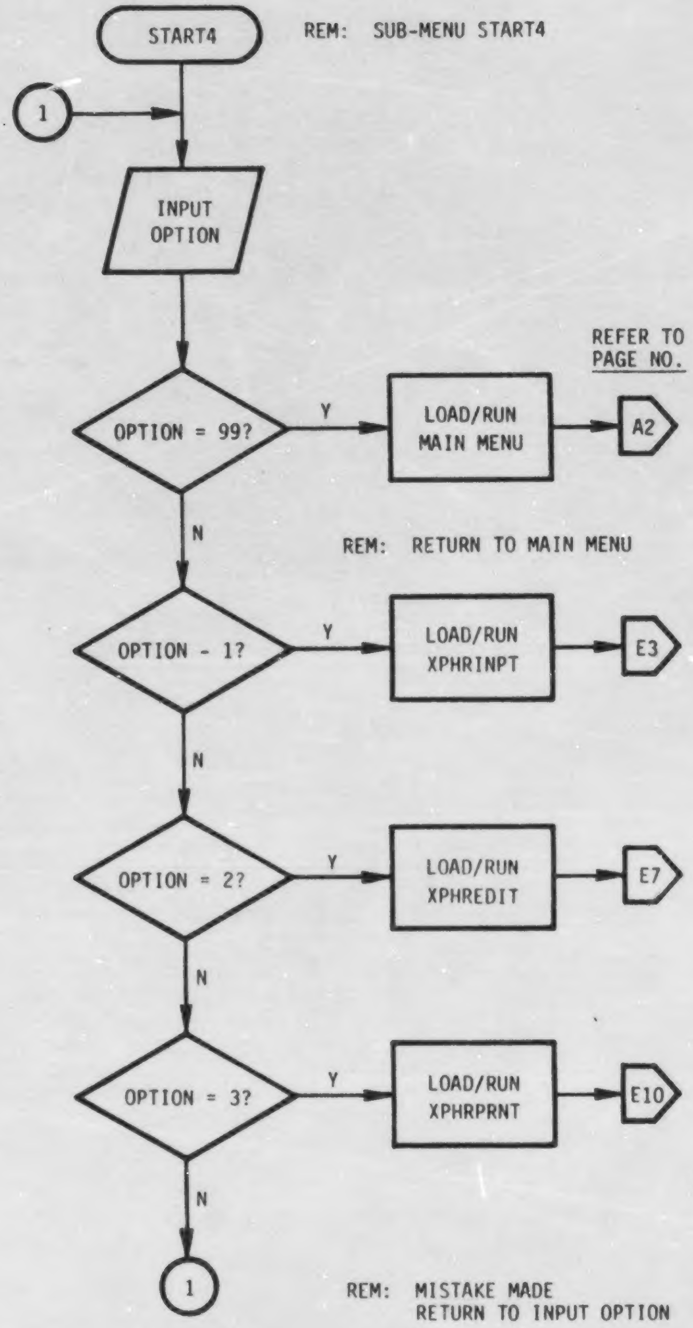
START4
REM: SUB-MENU START4
1
INPUT OPTION
REFER TO PAGE NO.
OPTION = 99?
Y
LOAD/RUN MAIN MENU
A2
N
REM: RETURN TO MAIN MENU
OPTION - 1?
Y
LOAD/RUN XPHRINPT
E3
N
OPTION = 2?
Y
LOAD/RUN XPHREDIT
E7
N
OPTION = 3?
Y
LOAD/RUN XPHRPRNT
E10
N
1
REM: MISTAKE MADE
RETURN TO INPUT OPTION

Figure 10. Flow Chart For Phrase Substitution Subsystem

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     PHRASE IS ALREADY ON FILE
     ENTER:
     (1)  TO DELETE PHRASE
     (2)  TO CHANGE SUBSTITUTES
     (3)  TO LEAVE PHRASE UNCHANGED
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you wish to delete the phrase enter "1". If you wish to change the substitutes then enter "2". If you wish to leave the phrase as it is then enter "3". If the phrase just entered is not on the file then the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER FIRST SUBSTITUTE OR C/R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the first substitute or press carriage return if the substitute is already on file and you do not wish to change it. Next, the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER SECOND SUBSTITUTE OR C/R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Enter the second substitute or press carriage return if you do not want to change or add a second substitute.

Edit. To edit a phrase substitution file that already exists, enter "2" as the option number for the subsystem menu. The edit program works exactly like the input program except that the phrase substitution file must already exist. You can add or delete phrases and you can change substitutes with this program.

Print To print a phrase substitution file enter "3" as the option number for the subsystem menu. This print program works exactly the same way as the program to print a word substitution file.

TEXT ANALYSIS SUBSYSTEM. To start the text analysis subsystem enter "5" as the option number in the Master Menu. The following menu will then be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
               * * * TEXT ANALYSIS SUBSYSTEM MENU * * *

     OPTIONS

        1      -    LOAD WORD AND PHRASE ANALYSIS PROGRAM
        2      -    ** FOR FUTURE USE **
        3      -    ** FOR FUTURE USE **

       99      -    ** Return to Master Menu **

     ENTER OPTION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The flow of control for the submenu START5 is shown in figure 11. The computer program for the text analysis subsystem is in appendix F. Additional details of the flow of control for submenu START5, relating to XXANAL04 are shown in figures 12, 13, 14, and 15.

If you wish to analyze a text passage or file then enter "1", otherwise enter "99". Options "2" and "3" are for future expansion of this system. After entering "1" the following message will be displayed:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Program        READABILITY PROGRAM (ANALYSIS)
  Version      2.0 - 5/27/80
  Abstract

               This program will compute a readability index for
               a passage of text.  The index calculated is based
               on the KINCAID INDEX.  This program is a modification
               of the GM SERVICE RESEARCH READABILITY PROGRAM.
  Touch 'RETURN(EXEC)' to Continue with Program...
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

This is just an identification message and no input is necessary. Press C/R to continue.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE FOR WORD SUBSTITUTION LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the device address where the word substitution file is stored. A programmer can supply you with the correct address.

Now you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE FOR PHRASE SUBSTITUTION LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the device address where the phrase substitution file is stored.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE FOR THE TEXT
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the device address where the text to be analyzed is stored.

Next you will be asked:

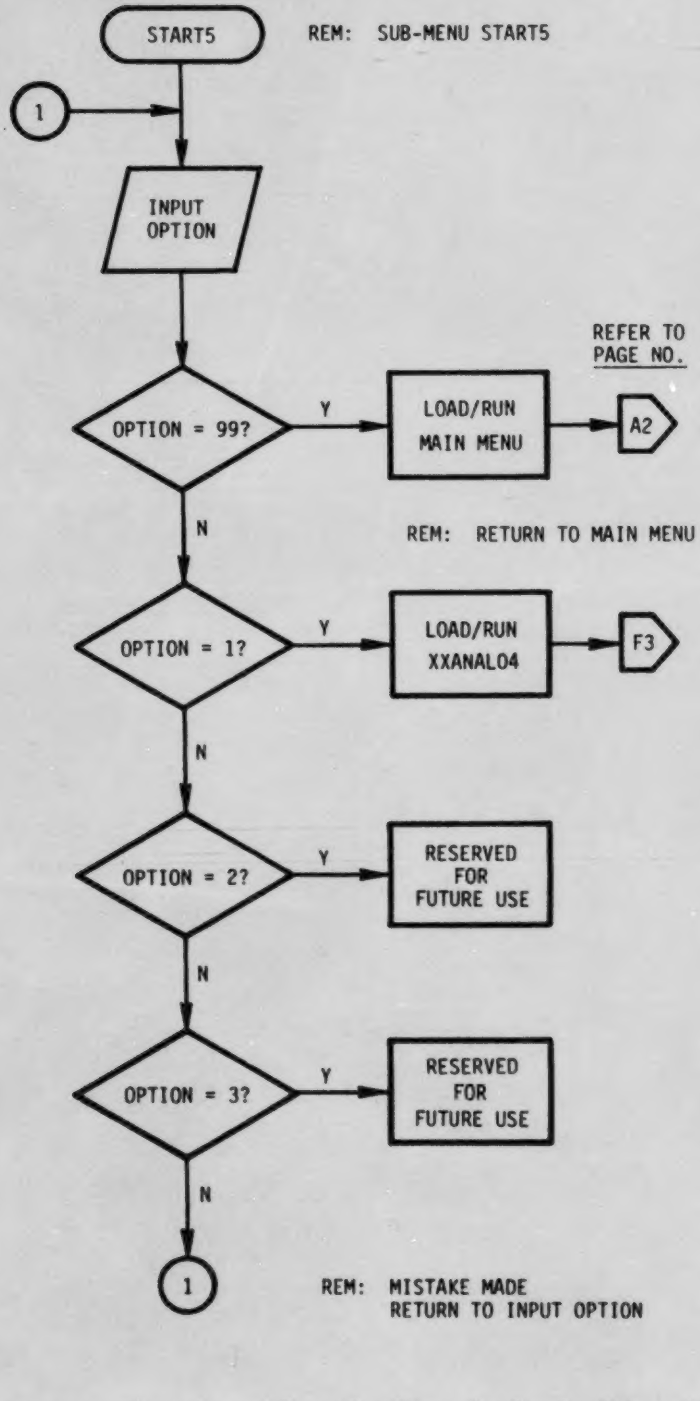
START5
REM: SUB-MENU START5
1
INPUT OPTION
REFER TO PAGE NO.
OPTION = 99?
Y
LOAD/RUN MAIN MENU
A2
N
REM: RETURN TO MAIN MENU
OPTION = 1?
Y
LOAD/RUN XXANAL04
F3
N
OPTION = 2?
Y
RESERVED FOR FUTURE USE
N
OPTION = 3?
Y
RESERVED FOR FUTURE USE
N
1
REM: MISTAKE MADE
RETURN TO INPUT OPTION

Figure 11. Flow Chart For the Text Analysis Subsystem

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE FILE NAME OF WORD SUBSTITUTION LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the file name of the word substitution file. A programmer can give you the correct name.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE FILE NAME OF THE PHRASE SUBSTITUTION LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the file name of the phrase substitution file.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE FILE NAME OF THE BLOCK POINTER FILE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the file name of the block pointer file that was created at the same time as the word substitution file that you are currently using. A programmer can give you the correct file name.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     DO YOU WISH TO USE A SUPPLEMENTARY WORD LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in "Y" for yes or "N" for no. If you key in "Y" then you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE OF SUPPLEMENTARY WORD LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the device address where the supplementary word file is stored.

Then you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE FILE NAME OF SUPPLEMENTARY WORD LIST
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the file name of the supplementary word file you want to use. A programmer can give you a list of the different supplementary word lists that are available.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER FILE NAME OF TEXT TO ANALYZE OR END TO STOP
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in the file name of the text in the form "SSSSnnnn"; where "SSSS" is the file identifier and "nnnn" is the sequence number of the passage within the file. If you want to analyze a whole file then the sequence number you should enter is "0000". Be sure to use zeros and not the letter O since the machine is expecting a number.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     DO YOU WANT TO TITLE THIS PASSAGE?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should key in "Y" for yes or "N" for no. If you enter "Y" then you will be allowed to enter a title of up to 40 characters. This title will then be printed at the top of the first page of the analysis.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     DO YOU WANT TO PRINT ONE OR TWO SUGGESTED SUBSTITUTES?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should enter either "1" or "2". This tells the machine how many substitutes to print when it finds a word or a phrase in the text that has suggested substitutes.

Next you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     IS TEXT IN UPPER CASE LETTERS ONLY?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should enter "Y" for yes or "N" for no.

Finally you will be asked:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ENTER THE DEVICE FOR PRINTING THE ANALYSIS RESULTS
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

You should enter the device address of the printer you wish to use to print the analyzed text. The computer will now analyze and print the text you requested.

Figures 12, 13, 14 and 15 contain the logic of additional features of the CRES including phrase analysis, word analysis, and printing. These figures contain the details of part of the text program contained in figure 11 which depicts the analysis subsystem. The computer program for this subsystem is in appendix F.

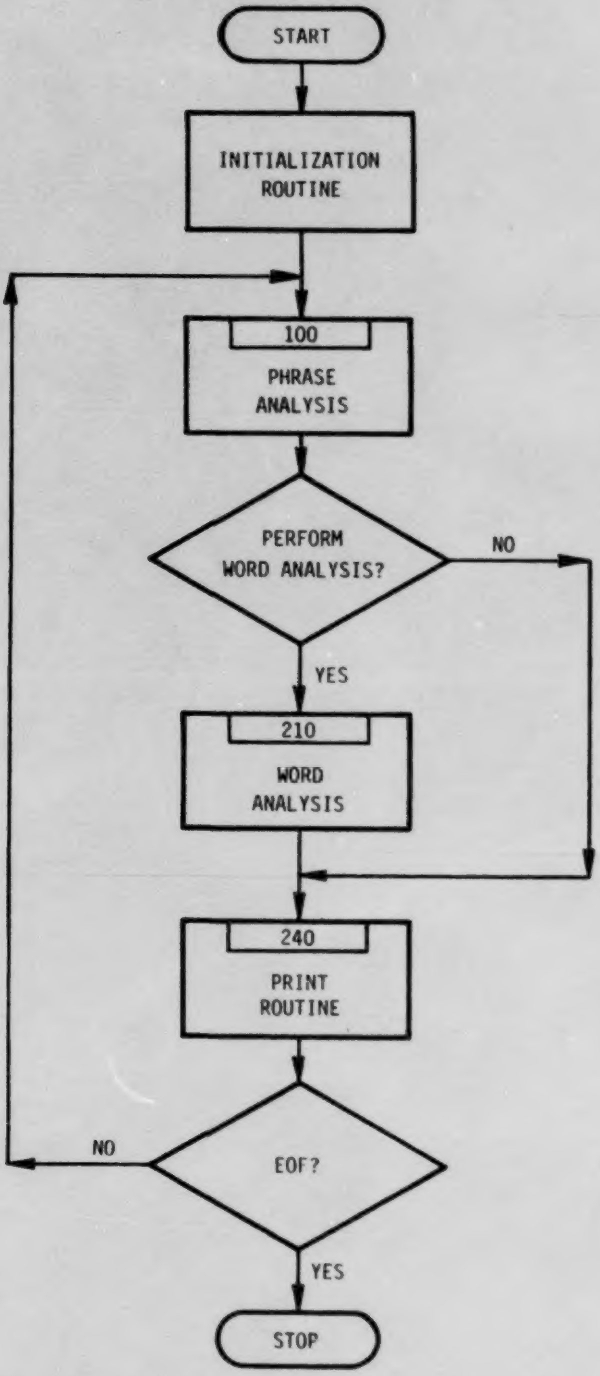
START
INITIALIZATION ROUTINE
100
PHRASE ANALYSIS
PERFORM WORD ANALYSIS?
NO
YES
210
WORD ANALYSIS
240
PRINT ROUTINE
EOF?
NO
YES
STOP

Figure 12. Flow Chart For Program XXANAL04

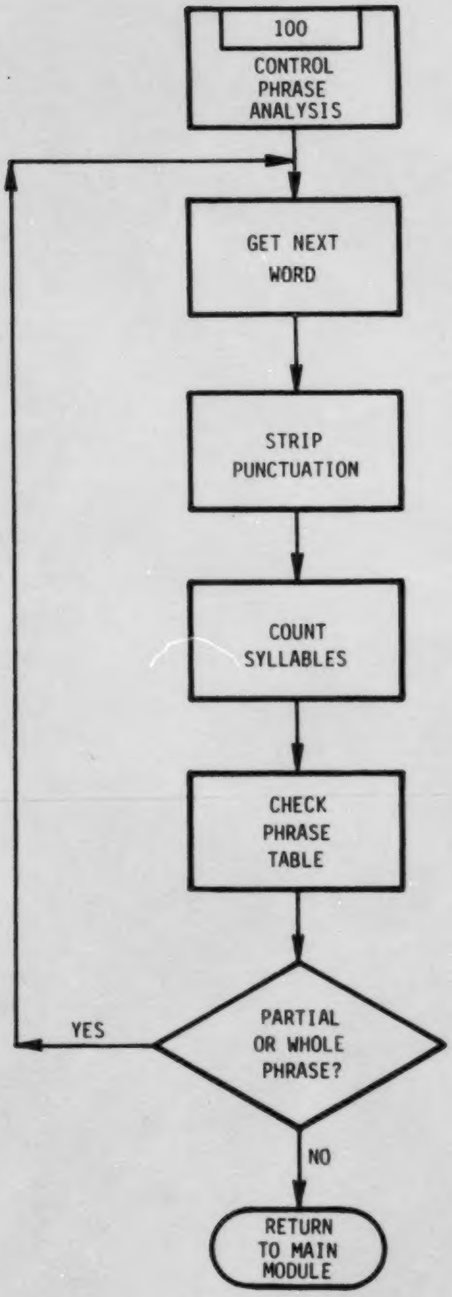
100
CONTROL PHRASE ANALYSIS
GET NEXT WORD
STRIP PUNCTUATION
COUNT SYLLABLES
CHECK PHRASE TABLE
PARTIAL OR WHOLE PHRASE?
YES
NO
RETURN TO MAIN MODULE

Figure 13. Flow Chart for The Phrase Analysis Subroutine of XXANAL04

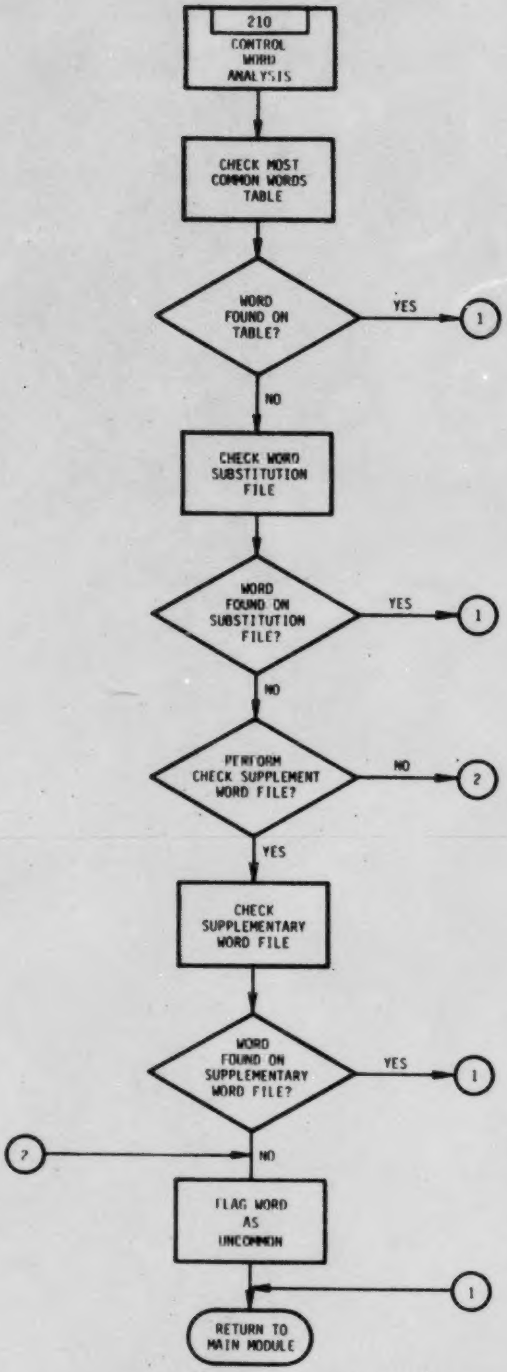
210
CONTROL WORD ANALYSIS
CHECK MOST COMMON WORDS TABLE
WORD FOUND ON TABLE?
YES
1
NO
CHECK WORD SUBSTITUTION FILE
WORD FOUND ON SUBSTITUTION FILE?
YES
1
NO
PERFORM CHECK SUPPLEMENT WORD FILE?
NO
2
YES
CHECK SUPPLEMENTARY WORD FILE
WORD FOUND ON SUPPLEMENTARY WORD FILE?
YES
1
2
NO
FLAG WORD AS UNCOMMON
1
RETURN TO MAIN MODULE

Figure 14. Flow Chart For The Word Analysis Subroutine of XXANAL04

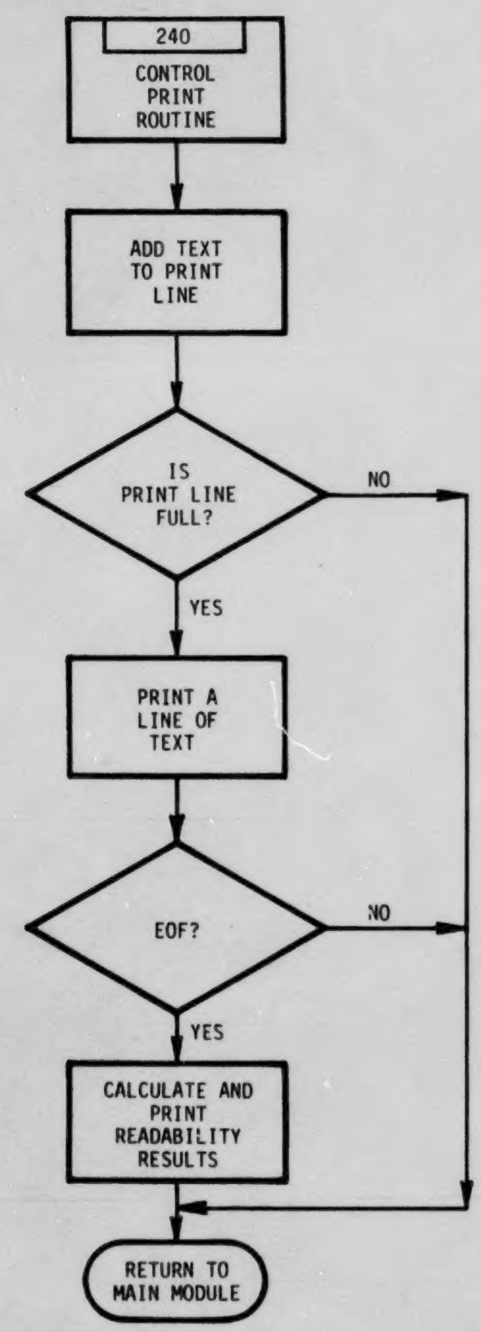
240
CONTROL PRINT ROUTINE
ADD TEXT TO PRINT LINE
IS PRINT LINE FULL?
NO
YES
PRINT A LINE OF TEXT
EOF?
NO
YES
CALCULATE AND PRINT READABILITY RESULTS
RETURN TO MAIN MODULE

Figure 15. Flow Chart For The Print Subroutine of XXANAL04

REFERENCES

General Accounting Office. Improved Management of Maintenance Manuals Needed in DOD. 1979. Washington, D.C.

General Accounting Office. A Need to Address Illiteracy Problems in the Military Services. 1977. Washington, D.C.

Keeler, F. L. An Automated Publishing System for the Naval Education and Training Command. TAEG Report No. 50, 1977. Training Analysis and Evaluation Group, Orlando, FL 32813. (ADA047648)

Kincaid, J. P., Aagard, J. A., and O'Hara, J. W. Development and Test of a Computer Readability Editing System (CRES). TAEG Report No. 83, 1980. Training Analysis and Evaluation Group, Orlando, FL. 32813.

Military Publications: Improving the Readability of Army Publications. Circular No. 310-9, 15 December 1978. Headquarters Department of the Army.

APPENDIX A

MASTER MENU

The LOAD RUN command, as described in section III, loads the program START into the main memory and initiates the execution of the readability system. The program START is the master menu of the system and provides the user the means of requesting one of the five subsystems. When a specific subsystem is selected by option number, a program which is the menu for the subsystem is loaded into memory and executed. The master menu also provides an option for terminating the execution of the readability system.

```
0010 %## READABILITY SYSTEM MASTER MENU PROGRAM (START) VERSION 2.0
0020 DIM C$(7)13,R$(7)13,E$(7)13,S$(7)13,N$12,N2$2,N3$3,L$80,Z1$1
0025 DIM B1$(3)42,C1$(28)13,L1$80,N1$1,U$(8)60
0026 B1$(1)=HEX(200C200C4F202020200A4F0A2008080808080808080808200A4F0A202020204
     F0C200C2008080808084F)
0027 B1$(2)=HEX(4F0C200C20204F20200A200A4F08080808080808080808084F0A200A20204F202
     00C200C4F08080808084F)
0028 B1$(3)=HEX(200C4F0C202020204F0A200A2008080808080808080808200A200A4F2020202
     00C4F0C2008080808084F)
0029 U$(1)=HEX(20202020202020202020202020202020202020202020202020202020202020
     20202020202020202020A0A0A0A02D2D2D2D3D3D2D2D2D2DA0A0A0A020)
0030 U$(2)=HEX(20202020202020202020208B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B2020
     2020202020202020202081818181818181818181818181818181818181)
0031 U$(3)=HEX(8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B2020202020202020202020
     20202020202020202020818120202020208382202020202020202020202020)
0032 U$(4)=HEX(2020202020202020208181202020202081812020202020202020202020202020
     20202020202020202020818120202020202020202020202020202020202020)
0033 U$(5)=HEX(2020202020202020208181202020208181202020202020202020202020202020
     2020202020202020208181202020202020202020202020202020202020202020)
0034 U$(6)=HEX(2020202020202020202020818120818120202020202020202020202020202020
     202020202020202081812020202020202020202020202020202020202020)
0035 U$(7)=HEX(20202020202020202020208181818181818181818181818181818181818181
     8181818181818181818181202020202020202020202020202020202020)
0036 U$(8)=HEX(20202020202020202020202081818181818181818181818181818181818181
     8181818181818181818181202020202020202020202020202020202020)
0049 Z1$="Y"
0050 N$=HEX(818181818181818181818181)
0051 N2$=HEX(8181)
0052 N3$=HEX(818181)
0053 C$(1),C$(7)=N$
0054 C$(2),C$(3),C$(4),C$(5),C$(6)=N3$ & "          "
0055 R$(1),R$(4)=STR(N$,1,9) & "    "
0056 R$(2),R$(3)=N3$ & "    " & N3$ & "   "
0057 R$(5)=N3$ & "      " & N2$ & "  "
0058 R$(6)=N3$ & "       " & N2$ & " "
0059 R$(7)=N3$ & "        " & N2$
0060 E$(1),E$(7)=N$
0061 E$(2),E$(3),E$(5),E$(6)=N3$ & "          "
0062 E$(4)=STR(N$,1,9)
0063 S$(1),S$(4),S$(7)=N$
0064 S$(2),S$(3)= N3$ & "          "
0065 S$(5),S$(6)= "          " & N3$
0066 IF Z1$="Y" THEN 68
0067 RETURN

-0068 L$="CRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRESCRES
```

```
      CRESCRESCRES"
    : C1$()= STR(C$()) & STR(R$()) & STR(E$()) & STR(S$())
 0069 SELECT PRINT 005(160)
 0070 R=INT(10*RND(1))+1
 0071 ON R GOSUB 2000,2200,3000,3500,4000,4400,4800,3000,3500,4000
 0072 FOR I=1 TO 3000
    : NEXT I
 0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
    : PRINTUSING 580
    : PRINTUSING 595
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633,634,635
 0390 IF O=99 THEN 1390
 0400 O=0
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %             * * *  READABILITY SYSTEM MASTER MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1      -  LOAD TEXT FILE SUBSYSTEM
-0565 %     2      -  LOAD WORD SUBSTITUTION SUBSYSTEM
-0575 %     3      -  LOAD SUPPLEMENTARY WORD SUBSYSTEM
-0580 %     4      -  LOAD PHRASE SUBSTITUTION SUBSYSTEM
-0595 %     5      -  LOAD TEXT ANALYSIS SUBSYSTEM
-0630 %    99      -  ** End of Session  **
-0631 LOAD DC T#0, "START1"

-0632 LOAD DC T#0, "START2"

-0633 LOAD DC T#0, "START3"

-0634 LOAD DC T#0, "START4"

-0635 LOAD DC T#0, "START5"
```

```
-1390 A,A1=0
 1400 PRINT HEX(0306)
 1410 A2=79
    : A4=0
 1420 A3=22
    : A5=0
-1430 PRINT AT(A,A1);HEX(81)
-1440 A1=A1+1
 1450 IF A1[A2THEN 1430
 1460 PRINT AT(A,A1);" "
 1470 A2=A2-1
-1480 A=A+1
 1490 PRINT AT(A,A1);HEX(83)
 1500 IF A[A3THEN 1480
 1510 PRINT AT(A,A1);" "
 1520 A3=A3-1
-1530 A1=A1-1
 1540 PRINT AT(A,A1);HEX(81)
 1550 IF A1]A5THEN 1530
 1560 PRINT AT(A,A1);" "
 1570 A5=A5+1
-1580 A=A-1
 1590 PRINT AT(A,A1);HEX(82)
 1600 IF A]A4THEN 1580
 1610 PRINT AT(A,A1);" "
 1620 A4=A4+1
 1630 IF A2[68THEN 1650
 1640 GOTO 1440

-1650 PRINT AT(10,10,60);" "
    : PRINT AT(10,25);"H A V E   A   N I C E   D A Y!!!"
 1655 PRINT AT(11,10,60);" "
 1660 PRINT AT(11,25);"H A V E   A   N I C E   D A Y!!!"
 1670 GOTO 9999

-2000 DEFFN '0
 2010 PRINT HEX(03)
 2020 PRINT AT(25,0)
 2025 FOR J= 1 TO 5
 2030 FOR I=1 TO 7
 2040 PRINT TAB(10);C$(I);TAB(24);R$(I);TAB(38);E$(I);TAB(52);S$(I)
 2045 FOR K=1 TO 100
    : NEXT K
 2050 NEXT I
 2060 FOR I=1 TO 18
```

```
2070 PRINT
2075 FOR K=1 TO 100
   : NEXT K
2080 NEXT I
2090 NEXT J
2108 PRINT AT(25,0)
2110 FOR I=1 TO 7
2120 PRINT TAB(10);C$(I);TAB(24);R$(I);TAB(38);E$(I);TAB(52);S$(I)
2122 FOR K=1 TO 100
   : NEXT K
2125 NEXT I
2145 FOR I=1 TO 8
   : PRINT
2146 FOR K=1 TO 100
   : NEXT K
2147 NEXT I
2150 RETURN

-2200 DEFFN '1
2205 PRINT HEX(06)
2210 PRINT HEX(03)
2220 FOR K=1 TO 13 STEP 2
2230 PRINT AT(8,0)
2235 FOR I=1 TO 7
2240 PRINT TAB(80-K);STR(C$(I),1,K)
2250 NEXT I
2260 NEXT K
2270 FOR K=14 TO 70 STEP 2
2280 PRINT AT(8,0)
2285 FOR I=1 TO 7
2290 PRINT TAB(80-K);STR(C$(I));" "
2300 NEXT I
2310 NEXT K
2370 FOR K=1 TO 13 STEP 2
2380 PRINT AT(8,0)
2390 FOR I=1 TO 7
2400 PRINT AT(7+I,80-K);STR(R$(I),1,K)
2410 NEXT I
2420 NEXT K
2430 FOR K=14 TO 56 STEP 2
2440 PRINT AT(8,0)
2450 FOR I=1 TO 7
2460 PRINT AT(7+I,80-K);STR(R$(I));" "
2470 NEXT I
2480 NEXT K
2490 FOR K=1 TO 13 STEP 2
```

```
2500 PRINT AT(8,0)
2510 FOR I=1 TO 7
2520 PRINT AT(7+I,80-K);STR(E$(I),1,K)
2530 NEXT I
2540 NEXT K
2550 FOR K=14 TO 42 STEP 2
2560 PRINT AT(8,0)
2570 FOR I=1 TO 7
2580 PRINT AT(7+I,80-K);STR(E$(I));" "
2590 NEXT I
2600 NEXT K
2610 FOR K=1 TO 13 STEP 2
2620 PRINT AT(8,0)
2630 FOR I=1 TO 7
2640 PRINT AT(7+I,80-K);STR(S$(I),1,K)
2650 NEXT I
2660 NEXT K
2670 FOR K=14 TO 28 STEP 2
2680 PRINT AT(8,0)
2690 FOR I=1 TO 7
2700 PRINT AT(7+I,80-K);STR(S$(I));" "
2710 NEXT I
2720 NEXT K
2730 RETURN

-3000 DEFFN '2
3002 PRINT HEX(06)
3005 FOR K=1 TO 3
3010 PRINT HEX(03)
3020 FOR I=0 TO 40
3030 PRINT AT(12,0);TAB(I);HEX(82);
3040 PRINT AT(12,79-I,2);HEX(83)
3050 NEXT I
3190 FOR I=1 TO 11
3195 PRINT AT(13-I,40-I);HEX(2A);
3200 PRINT AT(13-I,40-3*I);HEX(2A);
3205 PRINT AT(13-I,40-2*I);HEX(2A);
3210 PRINT AT(13-I,40);HEX(2A);
3215 PRINT AT(13-I,40+2*I);HEX(2A);
3220 PRINT AT(13-I,40+3*I);HEX(2A);
3224 PRINT AT(13-I,40+I);HEX(2A);
3226 PRINT AT(11+I,40-I);HEX(2A);
3230 PRINT AT(11+I,40-3*I);HEX(2A);
3235 PRINT AT(11+I,40-2*I);HEX(2A);
3240 PRINT AT(11+I,40);HEX(2A);
3245 PRINT AT(11+I,40+2*I);HEX(2A);
```

```
 3250 PRINT AT(11+I,40+3*I);HEX(2A);
 3260 PRINT AT(11+I,40+I);HEX(2A);
 3270 PRINT AT(12,40-3*I);HEX(2A);
 3275 PRINT AT(12,40+3*I);HEX(2A);
 3280 NEXT I
 3290 NEXT K
 3300 FOR I=1 TO 7
 3310 PRINT AT(8+I,14);STR(C$(I));" ";STR(R$(I));" ";STR(E$(I));" ";STR(S$(I))
      ;
 3320 NEXT I
 3330 RETURN

-3500 DEFFN '3
 3510 PRINT HEX(03)
 3520 PRINT HEX(06)
 3525 $TRAN(B1$(),HEX(434F))R
 3530 FOR I=1 TO 15
 3540 FOR J=1 TO 3
 3550 PRINT AT(12,15);B1$(J)
 3560 FOR K=1 TO 100
    : NEXT K
 3570 NEXT J
 3580 NEXT I
 3590 FOR I=1 TO 7
 3600 PRINT AT(8+I,12);STR(C$(I),1,12)
 3610 NEXT I
 3615 $TRAN(B1$(),HEX(5243))R
 3620 FOR I=1 TO 15
 3630 FOR J=1 TO 3
 3640 PRINT AT(12,29);B1$(J);
 3650 FOR K=1 TO 100
    : NEXT K
 3660 NEXT J
 3670 NEXT I
 3680 FOR I=1 TO 7
 3690 PRINT AT(8+I,26);STR(R$(I),1,12)
 3700 NEXT I
 3705 $TRAN(B1$(),HEX(4552))R
 3710 FOR I=1 TO 15
 3720 FOR J=1 TO 3
 3722 PRINT
 3730 PRINT AT(12,42);B1$(J);
 3740 FOR K=1 TO 100
    : NEXT K
 3750 NEXT J
 3760 NEXT I
```

```
3770 FOR I=1 TO 7
3780 PRINT AT(8+I,40);STR(E$(I),1,12)
3790 NEXT I
3795 $TRAN(B1$(),HEX(5345))R
3800 FOR I=1 TO 15
3810 FOR J=1 TO 3
3820 PRINT AT(12,56);B1$(J);
3830 FOR K=1 TO 100
   : NEXT K
3840 NEXT J
3850 NEXT I
3860 FOR I=1 TO 7
3870 PRINT AT(8+I,54);STR(S$(I),1,12)
3880 NEXT I
3885 $TRAN(B1$(),HEX(4F53))R
3890 RETURN

-4000 DEFFN '4
4005 SELECT PRINT 405(160)
4010 PRINT HEX(0306)
4015 S=MOD(INT(100*RND(1))+1,4)+1
4017 FOR J=1 TO 4
4018 FOR I=1 TO 4
-4020 Q=MOD(INT(100*RND(1))+1,4)+1
4030 IF Q=S THEN 4020
4040 GOSUB 4130
4050 ON Q GOSUB 4180,4220,4260,4300
4060 S=Q
4070 FOR K=1 TO 2000
   : NEXT K
4080 NEXT I
4085 GOSUB 4130
4090 FOR L=1 TO 7
4100 PRINT AT(8+L,(J*14)-2);STR(C1$(L+(J*7)-7),1,13)
4110 NEXT L
4120 GOTO 4340

-4130 FOR B=1 TO 7
4140 PRINT AT(B-1,0,79)
4150 PRINT AT(B+16,0,79)
4160 NEXT B
4170 RETURN

-4180 FOR L=1 TO 7
4190 PRINT AT(L-1,0);STR(C1$(L+(J*7)-7),1,13)
4200 NEXT L
```

```
 4210 RETURN

-4220 FOR L=1 TO 7
 4230 PRINT AT(L-1,66);STR(C1$(L+(J*7)-7),1,13)
 4240 NEXT L
 4250 RETURN

-4260 FOR L=1 TO 7
 4270 PRINT AT(L+16,0);STR(C1$(L+(J*7)-7),1,13);
 4280 NEXT L
 4290 RETURN

-4300 FOR L=1 TO 7
 4310 PRINT AT(L+16,66);STR(C1$(L+(J*7)-7),1,13);
 4320 NEXT L
 4325 SELECT PRINT 005(160)
 4330 RETURN

-4340 NEXT J
 4350 RETURN

-4400 DEFFN '5
 4410 PRINT HEX(0306)
 4420 FOR I=1 TO 24
 4430 PRINT " ";STR(L$,1,78)
 4440 NEXT I
 4450 FOR J=1 TO 4
 4460 FOR I=1 TO 76 STEP 3
 4470 L1$=L$
 4480 STR(L1$,I,3)="   "
 4490 STR(L1$,77-I,3)="   "
 4500 PRINT " ";STR(L1$,1,78)
 4510 FOR K=1 TO 50
    : NEXT K
 4520 NEXT I
 4530 NEXT J
 4540 FOR I=1 TO 31 STEP 3
 4550 L1$=L$
 4560 STR(L1$,I,3)="   "
 4570 STR(L1$,77-I,3)="   "
 4580 PRINT " ";STR(L1$,1,78)
 4590 FOR K=1 TO 50
    : NEXT K
 4600 NEXT I
 4610 $TRAN(C$(),HEX(09202081))R
 4620 $TRAN(R$(),HEX(09202081))R
```

```
4630 $TRAN(E$(),HEX(09202081))R
4640 $TRAN(S$(),HEX(09202081))R
4650 FOR I=1 TO 7
4660 PRINT AT(8+I,14);C$(I);HEX(09);R$(I);HEX(09);E$(I);HEX(09);S$(I)
4670 NEXT I
4680 $TRAN(C$(),HEX(81202009))R
4690 $TRAN(R$(),HEX(81202009))R
4700 $TRAN(E$(),HEX(81202009))R
4710 $TRAN(S$(),HEX(81202009))R
4720 RETURN

4800 DEFFN '6
4810 PRINT HEX(0306)
4815 FOR K=1 TO 23
4820 FOR I=1 TO 8
4821 PRINT AT(I-1,0);STR(U$(I),61-K)
4822 NEXT I
4823 NEXT K
4824 FOR I=1 TO 13
4825 PRINT AT(2+I,13);HEX(2E);AT(2+I,13+I);HEX(2E)
4826 NEXT I
4827 J=1
4828 GOSUB 5000
4829 FOR I=1 TO 13
4830 PRINT AT(2+I,13);HEX(20);AT(2+I,13+I);HEX(20)
4831 NEXT I
4832 FOR L=1 TO 2
4833 FOR K=1 TO 14
4834 FOR I=1 TO 8
4835 PRINT AT(I-1,0);STR(U$(I),38-(14*(L-1))-K)
4836 NEXT I
4837 NEXT K
4838 FOR I=1 TO 13
4839 PRINT AT(2+I,27+(14*(L-1)));HEX(2E);AT(2+I,27+I+(14*(L-1)));HEX(2E)
4840 NEXT I
4841 J=L+1
4842 GOSUB 5000
4843 FOR I=1 TO 13
4844 PRINT AT(2+I,27+(14*(L-1)));HEX(20);AT(2+I,27+I+(14*(L-1)));HEX(20)
4845 NEXT I
4846 NEXT L
4847 FOR K=0 TO 8
4848 FOR I=1 TO 8
4849 PRINT AT(I-1,0);STR(U$(I),9-K)
4850 NEXT I
4851 NEXT K
```

```
 4852 FOR K=1 TO 5
 4853 FOR I=1 TO 8
 4854 PRINT AT(I-1,K);STR(U$(I))
 4855 NEXT I
 4856 NEXT K
 4857 FOR I=1 TO 13
 4858 PRINT AT(2+I,55);HEX(2E);AT(2+I,55+I);HEX(2E)
 4859 NEXT I
 4860 J=4
 4861 GOSUB 5000
 4862 FOR I=1 TO 13
 4863 PRINT AT(2+I,55);HEX(20);AT(2+I,55+I);HEX(20)
 4864 NEXT I
 4865 RETURN

-5000 N1$="."
 5020 FOR K=1 TO 3
 5030 FOR I=1 TO 50
 5040 T=MOD(INT(100*RND(1))+1,13)+1
 5050 T1=MOD(INT(100*RND(1))+1,7)+1
 5060 T2=MOD(INT(100*RND(1))+1,13)+1
 5070 T3=MOD(INT(100*RND(1))+1,7)+1
 5080 PRINT AT(15+T1,(11+T+J)+((J-1)*13));N1$
 5090 ON MOD(I,3) GOTO 5110,5110
 5100 PRINT AT(15+T3,(11+T2+J)+((J-1)*13));HEX(20)
-5110 NEXT I
 5120 $TRAN(N1$,HEX(3A2E813A))R
 5130 NEXT K
 5140 FOR M=1 TO 7
 5150 PRINT AT(15+M,(J*14)-1);STR(C1$(M+(J*7)-7),1,13);HEX(20)
 5160 NEXT M
 5180 RETURN

 9990 DEFFN '15
 9991 SCRATCH T"XCRESDSP"
    : SAVE DC T$()"XCRESDSP"
-9999 REM
```

APPENDIX B

TEXT FILE SUBSYSTEM

Program START1 is the menu for the text file subsystem. This program permits the user the means of selecting 1 of 3 programs. If option 1 is selected, then program XEXTINPT is loaded into central memory and the execution of the program is initiated. XEXTINPT is the program used for creating a new text passage which will later be analyzed by the programs XXANAL04 or XXANAL01. If the user selects option 2, then the program XEXTEDIT is loaded into central memory and executed. Program XEXTEDIT is used to edit existing text passages. When option 3 is selected by the user, the program XEXTPRNT is loaded into central memory and executed. Program XEXTPRNT allows the user to print a hard copy of a specific text passage. If the user selects option 4, then the master menu START is loaded and executed.

```
 0010 %## TEXT FILE SUBSYSTEM MENU PROGRAM  START1
-0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633
 0390 IF O=99 THEN 638
 0400 O=0
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %             * * *  TEXT FILE SUBSYSTEM MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1     -  LOAD TEXT INPUT PROGRAM
-0565 %     2     -  LOAD TEXT EDIT PROGRAM
-0575 %     3     -  LOAD TEXT PRINT PROGRAM
-0630 %    99     -  ** Return to Master Menu  **
-0631 LOAD DC T#0, "XEXTINPT"

-0632 LOAD DC T#0, "XEXTEDIT"

-0633 LOAD DC T#0, "XEXTPRNT"

-0638 LOAD DC T#0, "START" BEG 310
```

TEXT INPUT PROGRAM (XEXTINPT)

The text input program is designed to allow the user to input and edit a new text passage. Because the current version of the text editor can not handle more than one screen of text at a time, a special naming convention is used to allow the system to process text files that are longer than one screen (25 lines). Each text passage (one screen worth of text) should be given a filename in the form "SSSSnnnn" where "SSSS" is the text file identifier and "nnnn" is the sequence number of the text passage within the text file. Thus, three screens of text could make up one continuous text file by naming them "ABCD0100", "ABCD0200", and "ABCD0300". Then, if the user wishes to insert a passage between "ABCD0100" and "ABCD0200" he can name the new passage "ABCD0150". Whenever a passage is given a filename with a new file identifier, the text input program creates a new file called "SSSS0000" where "SSSS" is the new file identifier and "0000" is the number 0000. This file will consist of a table of all the filenames with the same file identifier. The table will be sorted by sequence number. Then, when the user wishes to print or analyze a whole text file he should enter the filename "SSSS0000" (where "SSSS" is the file identifier) when the system asks for the filename of the text file. If the user wishes to print or analyze just one passage then he should use the filename of that passage. For example, "ABCD0200" would print or analyze just the passage with filename "ABCD0200".

The text can be edited before saving it on disk by using the special function keys to move the cursor or insert or delete characters. The text is treated as one long character string. Therefore, when you insert or delete a character all of the text in the passage after that character is moved. Each text passage is saved on disk as 80 character lines blocked 3 per sector with 12 sectors used for each passage.

The following is a list of variables and their uses in this program:

A8$3 = Disk address where the text file will be stored

B$(25)80 = Output array for one screen of text

B1$(3)80 = One logical output record

B6$(2)2 = Locator array for MAT SEARCH of F2$()

F1$8 = The filename for the table of text passage filenames

F2$(500)8 = The table of text passage filenames

F3$(500)2 = Work area for MAT SORT of F2$()

F4$(500)2 = Locator array output for MAT SORT of F2$()

F5$(500)8 = Output array for MAT MOVE of F2$()

X1,X2 = Dimensions of text passage array

X7$(25)80 = Work area for passage in text editor subroutine

Z0$8 = Filename of text passage

Z6$1 = Yes/No switch for saving text on disk

```
 0001 REM   READABILITY SYSTEM TEXT INPUT PROGRAM  (XEXTINPT)
 0005 REM This program will allow the input and edit of a passage of text of s
      ize 25x80
 0040 DIM B$(25)80,Z0$8
 0050 DIM X7$(25)80,X1,X2
 0060 DIM Z6$1
 0070 DIM A8$3
 0071 DIM F5$(500)8,B6$(2)2,F1$8,F2$(500)8,F3$(500)2,F4$(500)2,B1$(3)80
 0072 X1=25
    : X2=80
 0075 PRINT HEX(03)
 0080 PRINT AT(7,2);"Enter the device for the Text storage";
    : LINPUT -A8$
 0090 SELECT #1 [A8$], #2 [A8$]
 2770 INIT(20)B$()
 2780 PRINT HEX(03)
 2790  PRINT "Enter Your Text Now (Touch Stmnt Nbr/FN Key to Exit)"
    : GOSUB '190 (2,24,25,80)
    : MAT COPY X7$() TO B$()
-2890 Z6$="Y"
    : PRINT HEX(03)
    : LINPUT "Do you Wish To Save This Text On Disk",-Z6$
    : ON POS("YyNn"=STR(Z6$,,1)) GOTO 2930,2930,2950,2950
    : ELSE PRINT HEX(07)
    : GOTO 2890

-2930 GOSUB '211
-2950 PRINT HEX(03)
 2960 PRINT AT(10,10)
    : FOR J=1 TO 5
    : PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
    : SELECT P3
    : PRINT HEX(03)
    : NEXT J
    : SELECT P0
    : LOAD DC T#0, "START1"

 6170 DEFFN' 211
-6190 PRINT HEX(0A)
 6200 PRINTUSING 6390," ";
 6210 INPUT Z0$
 6215 GOSUB '212
 6220 LIMITS T#1,STR(Z0$,1,8),L1,L2,L3,S
 6240 IF S]=0THEN 6310
-6250 DATA SAVE DC OPEN T#1,(STR(Z0$,1,8)),STR(Z0$,1,8)
-6260 FOR J=0 TO 21 STEP 3
```

```
 6261 FOR I=1 TO 3
 6262 B1$(I)=B$(J+I)
 6263 NEXT I
 6264 DATA SAVE DC #1,B1$()
 6265 B1$()=ALL(20)
 6266 NEXT J
 6267 B1$(1)=B$(25)
 6268 DATA SAVE DC #1,B1$()
 6270 DATA SAVE DC #1,END
 6280 DATA SAVE DC CLOSE #1
 6290 GOTO 6380

-6310 IF S]0THEN 6350
 6320 DATA SAVE DC OPEN T#1,(12),STR(Z0$,1,8)
 6330 GOTO 6260

-6350 PRINT HEX(0307)
 6351 PRINT "THE FILENAME YOU JUST ENTERED IS THE NAME OF AN ACTIVE FILE"
 6352 PRINT "ENTER Y IF YOU WISH TO SCRATCH THE FILE"
 6353 PRINT "ENTER N IF YOU WISH TO ENTER A DIFFERENT FILENAME"
 6354 INPUT Z6$
 6355 IF Z6$[]"Y" THEN 6190
 6356 SCRATCH T#1,STR(Z0$,1,8)
 6360 GOTO 6250

-6380 RETURN

-6390 % Enter The Filename of Text File to Save #
 6400 DEFFN '212
 6410 STR(F1$,1,4)=STR(Z0$,1,4)
 6420 STR(F1$,5,4)="0000"
 6430 LIMITS T#2, STR(F1$,1,8),L1,L2,L3,S
 6440 IF S=0 THEN 6500
 6450 IF S=2 THEN 6550
 6460 IF S=-2 THEN 6650
 6470 PRINT "INVALID TEXT FILE NAME"
 6480 PRINT "FILENAME",F1$,"ALREADY EXISTS AS A PROGRAM FILE"
 6490 STOP "END OF PROGRAM"
-6500 DATA SAVE DC OPEN T$#2,(20),F1$
-6510 F2$()=ALL (FF)
 6520 F2$(1)=Z0$
 6530 DATA LOAD DC OPEN T#2,STR(F1$,1,8)
 6540 GOTO 6670

-6550 DATA LOAD DC OPEN T#2,STR(F1$,1,8)
 6560 DATA LOAD DC #2,F2$()
 6570 MAT SEARCH F2$(), =Z0$ TO B6$() STEP 8
 6580 IF B6$(1)=HEX(0000) THEN 6600
 6590 GOTO 6690

-6600 F2$(500)=Z0$
 6610 MAT SORT F2$() TO F3$(),F4$()
 6620 MAT MOVE F2$(), F4$() TO F5$()
 6630 MAT COPY F5$() TO F2$()
 6640 GOTO 6670

-6650 DATA SAVE DC OPEN T$#2,(F1$), F1$
 6660 GOTO 6510

-6670 DBACKSPACE #2, BEG
    : DATA SAVE DC #2,F2$()
 6680 DATA SAVE DC #2,END
-6690 RETURN

 7000 %## PROGRAM ] INPUT UTILITIES    (INPT.SRC) VERSION 2.0  2200 T/VP/MVP
 7106 DEFFN'190(X3,X4,X5,X6)
 7108 DIM X3$(1)1,X0$(80)1
 7110 DIM X8$10,X9(5),X8$(1)1,X$(10)1
 7112 DIM X5$(1)64,X6$(1)2,X4$(1)64
 7114 DIM X8(2,6),X7(3)
 7116 X9(1)=1
 7118 X9(2)=X4-X3+1
 7120 X9(3)=1
 7122 INIT(09)X0$()
 7124 INIT(20)X8$()
 7126 INIT(20)X5$()
 7128 $GIO(010502160380420,X$())
 7130 $IF ON /0007134
 7132 $GIO(010302000380420,X$())
-7134 KEYIN X3$(1),7134,7134
 7136 X8=1
 7138 X9=1
 7140 GOSUB '191(X9(1),X9(2))
 7142 GOSUB '101(X9(3)+X3-1,1,0,0)
-7144 MAT COPY X7$()[(X8-1)*X2+X9,1]TO X$()[5,1]
-7146 KEYIN X3$(1),7156,7152
 7148 $GIOCURSOR(71054006426040081211425040081136145416081146,X$())
 7150 GOTO 7146

-7152 IF X3$(1)=HEX(7E)THEN 7288
 7154 GOSUB '192(X3$(1))
    : GOTO 7144
```

```
-7156 IF X3$(1)=HEX(E5)THEN 7276
 7158 IF X3$(1)=HEX(E6)THEN 7288
 7160 IF X3$(1)[]HEX(81)THEN 7164
 7162 INIT(20)X4$(),X5$()
    : GOTO 7144

-7164 IF X3$(1)[=HEX(7A)THEN 7168
 7166 $GIO/005(4007,X8$)
    : GOTO 7144

-7168 IF X3$(1)=HEX(0D)THEN 7188
 7170 IF X3$(1)=HEX(08)THEN 7238
 7172 $GIOPRINT/005(A200,X8$)X3$()[1,1]
 7174 MAT COPY X3$()TO X7$()[(X8-1)*X2+X9,1]
 7176 X9=X9+1
 7178 IF X9[=X6THEN 7182
 7180 IF X8]=X5THEN 7212
-7182 IF X9[]X2-5THEN 7186
 7184 $GIO/005(4007,X8$)
-7186 IF X9[X2+1THEN 7144
-7188 IF X8+1]=X1+1THEN 7212
 7190 IF X8+1]=X5+1THEN 7212
 7192 IF X9(3)+1[=X9(2)THEN 7202
 7194 X9(1)=X9(1)+1
 7196 GOSUB '191(X9(1),X9(2))
 7198 GOSUB '101(X9(3)+X3-1,1,0,0)
 7200 GOTO 7206

-7202 $GIOCRLF/005(400D400A,X8$)
 7204 X9(3)=X9(3)+1
-7206 X8=X8+1
 7208 X9=1
 7210 GOTO 7144

-7212 KEYIN X3$(1),7230,7214
    : GOTO 7212

-7214 IF X3$(1)=HEX(7E)THEN 7288
 7216 IF X3$(1)]HEX(0E)THEN 7236
 7218 IF X3$(1)=HEX(09)THEN 7236
 7220 IF X3$(1)=HEX(0A)THEN 7236
 7222 IF X3$(1)=HEX(02)THEN 7236
 7224 IF X3$(1)=HEX(03)THEN 7236
 7226 IF X3$(1)=HEX(00)THEN 7236
 7228 GOSUB '192(X3$(1))
```

```
    : GOTO 7144

-7230 IF X3$(1)=HEX(E6)THEN 7288
 7232 IF X3$(1)=HEX(08)THEN 7238
 7234 IF X3$(1)=HEX(E5)THEN 7276
-7236 $GIO/005(4007,X8$)
    : GOTO 7212

-7238 IF X9[]1THEN 7268
 7240 $GIORUBOUT/005(40204008,X8$)
 7242 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7244 IF X8=1THEN 7144
 7246 X8=X8-1
 7248 X9=X2
 7250 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7252 IF X9(3)[]1THEN 7262
 7254 X9(1)=X9(1)-1
 7256 GOSUB '191(X9(1),X9(2))
 7258 GOSUB '101(X9(3)+X3-1,X2,1,1)
 7260 GOTO 7144

-7262 X9(3)=X9(3)-1
 7264 $GIOUP/005(400CA20040204008,X8$)X0$()[1,X2-1]
 7266 GOTO 7144

-7268 X9=X9-1
 7270 $GIORUBOUT/005(400840204008,X8$)
 7272 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7274 GOTO 7144

-7276 X9=1
 7278 INIT(20)X0$()
 7280 MAT COPY X0$()[1,1]TO X7$()[(X8-1)*X2+1,X2]
 7282 $GIOCLRLINE/005(400DA200400D,X8$)X0$()[1,X2]
 7284 INIT(09)X0$()
 7286 GOTO 7144

-7288 RETURN

 7290 DEFFN'191(X0,X7)
 7292 GOSUB '101(X3,1,0,0)
 7294 FOR X=X0TO X7+X0-1
 7296 IF X=X7+X0-1THEN 7302
 7298 $GIOPRINT/005(A200400D400A,X8$)X7$()[(X-1)*X2+1,X2]
 7300 GOTO 7304
```

```
-7302 $GIOPRINT/005(A200,X8$)X7$()[(X-1)*X2+1,X2]
-7304 NEXT X
 7306 RETURN

 7308 DEFFN'192(X3$(1))
 7310 $GIO/005(4005,X8$)
 7312 IF X3$(1)[HEX(01)THEN 7316
 7314 IF X3$(1)[HEX(0F)THEN 7318
-7316 $GIO/005(40074006,X8$)
    : GOTO 7670

-7318 ON VAL(X3$(1))GOTO 7320,7334,7436,7504,7518,7538,7558,7572,7584,7600,761
      4,7628,7642,7656
-7320 X9(1)=(X9(1)+X9(2))-INT((X9(1)+X9(2))/X5)*X5
 7322 IF X9(1)+X9(2)[=X5+1THEN 7326
 7324 X9(1)=X5-X9(2)+1
-7326 GOSUB '191(X9(1),X9(2))
 7328 X8=X9(1)+X9(3)-1
 7330 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7332 GOTO 7670

-7334 KEYIN X3$(1),7334,7334
 7336 X=0
-7338 KEYIN X3$(1),7346,7340
    : GOTO 7338

-7340 IF X3$(1)=HEX(00)THEN 7344
 7342 IF X3$(1)[]HEX(02)THEN 7364
-7344 X=LEN(X5$(1))
    : GOTO 7356

-7346 IF X3$(1)=HEX(0D)THEN 7356
 7348 IF X3$(1)=HEX(E6)THEN 7364
 7350 X=X+1
 7352 MAT COPY X3$()TO X5$()[X,1]
 7354 IF X[63THEN 7338
-7356 MAT COPY X8$()TO X5$()[X+1,64-X]
 7358 X0=(X8-1)*X2+X9+1
 7360 MAT SEARCHX7$()[X0,(X1*X2)-X0+1],=X5$(1)TO X6$()
 7362 IF X6$(1)[]HEX(0000)THEN 7366
-7364 $GIOBELL/005(4007,X8$)
    : GOTO 7434

-7366 X0=256*VAL(STR(X6$(1),1,1))+VAL(STR(X6$(1),2,1))+X0-1
 7368 X8=INT((X0-1)/X2)+1
 7370 X9=X0-(X8-1)*X2
 7372 IF X8]=X9(1)+X9(2)THEN 7376
 7374 X9(3)=X8-X9(1)+1
    : GOTO 7386

-7376 X9(1)=X8-INT(X9(2)/2)
 7378 IF X9(1)+X9(2)[=X5+1THEN 7382
 7380 X9(1)=X5-X9(2)+1
-7382 X9(3)=X8-X9(1)+1
 7384 GOSUB '191(X9(1),X9(2))
-7386 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7388 IF X3$(1)[]HEX(00)THEN 7434
-7390 KEYIN X3$(1),7390,7390
 7392 X=0
-7394 KEYIN X3$(1),7400,7396
    : GOTO 7394

-7396 IF X3$(1)[]HEX(00)THEN 7364
 7398 X=LEN(X4$(1))
    : GOTO 7410

-7400 IF X3$(1)=HEX(OD)THEN 7410
 7402 IF X3$(1)=HEX(E6)THEN 7364
 7404 X=X+1
 7406 MAT COPY X3$()TO X4$()[X,1]
 7408 IF X[63THEN 7394
-7410 MAT COPY X8$()TO X4$()[X+1,64-X]
 7412 XO=(X8-1)*X2+X9
 7414 IF XO+LEN(X5$(1))[=X1*X2THEN 7418
 7416 STR(X5$(1),(X1*X2)-XO+1)=" "
-7418 X=LEN(X5$(1))
 7420 MAT COPY X7$()[XO+X,(X1*X2)-(XO+(X-1))]TO X7$()[XO,(X1*X2)-(XO+1)]
 7422 X=LEN(X4$(1))
 7424 X7=(X1*X2)-XO-(X-1)
 7426 MAT COPY -X7$()[XO,X7]TO -X7$()[XO+X,X7]
 7428 MAT COPY X4$()[1,X]TO X7$()[XO,X]
 7430 GOSUB '191(X9(1),X9(2))
 7432 GOSUB '101(X9(3)+X3-1,X9,0,0)
-7434 GOTO 7670

-7436 XO=(X8-1)*X2+X9
 7438 X8(1,1)=X9(1)
    : X8(1,2)=X9(2)
 7440 X8(1,3)=X9(3)
    : X8(1,4)=X8
 7442 X8(1,5)=X9
    : X8(1,6)=XO
```

```
-7444 KEYIN X3$(1),7444,7446
    : GOTO 7444

-7446 IF X3$(1)=HEX(03)THEN 7454
 7448 IF X3$(1)=HEX(E6)THEN 7464
 7450 IF X3$(1)=HEX(7E)THEN 7464
 7452 GOSUB '192(X3$(1))
    : GOTO 7444

-7454 X0=(X8-1)*X2+X9
 7456 X8(2,1)=X9(1)
    : X8(2,2)=X9(2)
 7458 X8(2,3)=X9(3)
    : X8(2,4)=X8
 7460 X8(2,5)=X9
    : X8(2,6)=X0
 7462 IF X8(2,6)-X8(1,6)]0THEN 7466
-7464 $GIOBELL/005(4007,X8$)
    : GOTO 7498

-7466 KEYIN X3$(1),7466,7468
    : GOTO 7466

-7468 IF X3$(1)=HEX(03)THEN 7476
 7470 IF X3$(1)=HEX(E6)THEN 7464
 7472 IF X3$(1)=HEX(7E)THEN 7464
 7474 GOSUB '192(X3$(1))
    : GOTO 7466

-7476 X7(1)=(X8-1)*X2+X9
 7478 X7(2)=X8(2,6)-X8(1,6)+1
 7480 X7=(X1*X2)-X7(1)-(X7(2)-1)
-7482 KEYIN X3$(1),7484,7464
    : GOTO 7482

-7484 IF X3$(1)="R"THEN 7492
 7486 IF X3$(1)="r"THEN 7492
 7488 MAT COPY -X7$()[X7(1),X7]TO -X7$()[X7(1)+X7(2),X7]
 7490 IF X7(1)[=X8(1,6)THEN 7496
-7492 MAT COPY X7$()[X8(1,6),X7(2)]TO X7$()[X7(1),X7(2)]
 7494 GOTO 7498

-7496 MAT COPY X7$()[X8(1,6)+X7(2),X7(2)]TO X7$()[X7(1),X7(2)]
-7498 GOSUB '191(X9(1),X9(2))
 7500 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7502 GOTO 7670
```

```
-7504 X9(3)=X9(2)
 7506 X9(1)=X5-X9(2)+1
 7508 X8=X5
 7510 X9=X6
 7512 GOSUB '191(X9(1),X9(2))
 7514 GOSUB '101(X4,X6,0,0)
 7516 GOTO 7670

-7518 IF X8]=X5THEN 7670
 7520 X8=X8+1
 7522 IF X9(3)[X9(2)THEN 7532
 7524 X9(1)=X9(1)+1
 7526 GOSUB '191(X9(1),X9(2))
 7528 GOSUB '101(X4,X9,0,0)
 7530 GOTO 7670

-7532 X9(3)=X9(3)+1
 7534 $GIOLF/005(400A,X8$)
 7536 GOTO 7670

-7538 IF X8[=1THEN 7670
 7540 X8=X8-1
 7542 IF X9(3)]1THEN 7552
 7544 X9(1)=X9(1)-1
 7546 GOSUB '191(X9(1),X9(2))
 7548 GOSUB '101(X3,X9,0,0)
 7550 GOTO 7670

-7552 X9(3)=X9(3)-1
 7554 $GIOUP/005(400C,X8$)
 7556 GOTO 7670

-7558 X9(3)=1
 7560 X9(1)=1
 7562 X8=1
 7564 X9=1
 7566 GOSUB '191(X9(1),X9(2))
 7568 GOSUB '101(X3,1,0,0)
 7570 GOTO 7670

-7572 X3$(1)=HEX(20)
 7574 X0=(X8-1)*X2+X9
 7576 MAT COPY X3$()[1,1]TO X7$()[X0,(X1*X2)-X0+1]
 7578 GOSUB '191(X9(1),X9(2))
 7580 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7582 GOTO 7670

-7584 X0=(X8-1)*X2+X9
 7586 IF X0[]X1*X2THEN 7592
 7588 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7590 GOTO 7594

-7592 MAT COPY X7$()[X0+1,(X1*X2)-X0]TO X7$()[X0,(X1*X2)-X0+1]
-7594 GOSUB '191(X9(1),X9(2))
 7596 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7598 GOTO 7670

-7600 X0=(X8-1)*X2+X9
 7602 X7=(X1*X2)-X0
 7604 MAT COPY -X7$()[X0,X7]TO -X7$()[X0+1,X7]
 7606 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7608 GOSUB '191(X9(1),X9(2))
 7610 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7612 GOTO 7670

-7614 IF X9+5[=X2THEN 7622
 7616 X9=1
 7618 $GIOCR/005(400D,X8$)
 7620 GOTO 7670

-7622 X9=X9+5
 7624 $GIORIGHT/005(40094009400940094009,X8$)
 7626 GOTO 7670

-7628 IF X9+1[=X2THEN 7636
 7630 X9=1
 7632 $GIOCR/005(400D,X8$)
 7634 GOTO 7670

-7636 X9=X9+1
 7638 $GIORIGHT/005(4009,X8$)
 7640 GOTO 7670

-7642 IF X9-1]=1THEN 7650
 7644 X9=X2
 7646 $GIOCRTAB/005(400DA200,X8$)X0$()[1,X2-1]
 7648 GOTO 7670

-7650 X9=X9-1
 7652 $GIOLEFT/005(4008,X8$)
 7654 GOTO 7670
-7656 IF X9-5]=1THEN 7664
 7658 X9=X2
 7660 $GIOCRTAB/005(400DA200,X8$)X0$()[1,X2-1]
 7662 GOTO 7670

-7664 X9=X9-5
 7666 $GIOLEFT/005(40084008400840084008,X8$)
 7668 GOTO 7670

-7670 RETURN

 7672 DEFFN'101(Q1,Q2,Q3,Q4)
 7674 DIM Q1$(24)1,Q2$(80)1,Q3$(80)1,Q4$(10)1
 7676 INIT(0A)Q1$()
 7678 INIT(09)Q2$()
 7680 INIT(20)Q3$()
 7682 Q1$(1)=HEX(01)
 7684 Q2$(1)=HEX(0D)
 7686 IF Q3=0THEN 7708
 7688 IF Q4[]0THEN 7692
 7690 Q4=81-Q2
-7692 FOR Q5=1TO Q3
 7694 IF Q5[]1THEN 7700
 7696 $GIOROW/005(A200,Q4$())Q1$()[1,Q1]
 7698 GOTO 7702

-7700 $GIOROW/005(A200,Q4$())Q1$()[2,1]
-7702 $GIOCOL/005(A200,Q4$())Q2$()[1,Q2]
 7704 $GIOERS/005(A200,Q4$())Q3$()[1,Q4]
 7706 NEXT Q5
-7708 $GIOROW/005(A200,Q4$())Q1$()[1,Q1]
 7710 $GIOCOL/005(A200,Q4$())Q2$()[1,Q2]
 7712 RETURN

 7800 GOSUB ' 239(1)
    : GOSUB ' 239(2)
```

TEXT EDIT PROGRAM (XEXTEDIT)

The text edit program is designed to allow the user to edit a passage of text that is already on disk. This program works just like the text input program except that the text passage and the table of text passage filenames must already be on disk. If the text passage filename is not in the table of filenames then it will be added to the table.

The variables and their uses are the same as in the text input program with the addition of:

N9$1 = Yes/No switch for editing another passage

```
 0010 REM    READABILITY SYSTEM TEXT EDIT PROGRAM   (XEXTEDIT)
 1000 REM This program prints the text from the file on the screen,
 1010 REM allows the change, and saves the new text
 1020 DIM B$(25)80,Z0$8,F5$(500)8,B6$(2)2,N9$1,B1$(3)80
 1030 DIM X7$(25)80,X1,X2,A9$3,F1$8,F2$(500)8,F3$(500)2,F4$(500)2
 1060 X2=80
    : X1=25
 1065 F5$()=ALL (FF)
 1070 PRINT HEX(0703)
 1080 PRINT AT(9,2);"Enter the device for TEXT";
    : LINPUT -A9$
 1090 SELECT #1 [A9$], #2 [A9$]
-1100 GOSUB ' 210
    : PRINT HEX(03)
    : INPUT "DO YOU WISH TO EDIT ANOTHER PASSAGE",N9$
    : IF N9$="Y" OR N9$="y" THEN 1100
 1120 PRINT HEX(03)
    : FOR J=1 TO 5
    : PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
    : SELECT P3
    : PRINT HEX(03)
    : NEXT J
    : SELECT P0
    : LOAD DC T#0, "START1"

 1130 DEFFN'210
 1140 SELECT PRINT 005(80)
    : PRINT HEX(07)
 1150 PRINT HEX(03)
-1160 PRINT AT(3,1);
 1170 PRINTUSING 1400," ";
 1180 INPUT Z0$
    : IF Z0$="END" THEN 1390
 1190 LIMITS T#1,STR(Z0$,1,8),L1,L2,L3,S
 1200 IF S=2THEN 1240
 1210 PRINT HEX(0707)
 1220 PRINTUSING 1410
 1230 GOTO 1160

-1240 DATA LOAD DC OPEN T#1,STR(Z0$,1,8)
 1250 DATA LOAD DC #1,B$()
 1255 GOSUB '212
 1260 PRINT HEX(03);" You may modify this passage now or touch 'STMT NUMBER' o
      r 'FN' to Exit"
 1270 MAT COPY B$() TO X7$()
    : GOSUB '190(2,24,25,80)
    : MAT COPY X7$() TO B$()
 1280 DBACKSPACE #1, BEG
 1290 FOR J=0 TO 21 STEP 3
 1300 FOR I=1 TO 3
 1310 B1$(I)=B$(J+I)
 1320 NEXT I
 1330 DATA SAVE DC #1,B1$()
 1340 B1$()=ALL(20)
 1350 NEXT J
 1360 B1$(1)=B$(25)
 1370 DATA SAVE DC #1,B1$()
 1380 DATA SAVE DC #1,END
-1390 RETURN

-1400 % Enter The Filename of Text File To Edit Or END to stop #
-1410 % INVALID FILENAME ....  RE-ENTER
 1500 DEFFN '212
 1510 STR(F1$,1,4)=STR(Z0$,1,4)
 1520 STR(F1$,5,4)="0000"
 1530 LIMITS T#2, STR(F1$,1,8),L1,L2,L3,S
 1540 IF S=0 THEN 1600
 1550 IF S=2 THEN 1640
 1560 IF S=-2 THEN 1710
 1570 PRINT "INVALID TEXT FILE NAME"
 1580 PRINT "FILENAME",F1$,"ALREADY EXISTS AS A PROGRAM FILE"
 1590 STOP "END OF PROGRAM"
-1600 DATA SAVE DC OPEN T$#2,(20),F1$
-1610 F2$()=ALL (FF)
 1620 F2$(1)=Z0$
 1630 DATA LOAD DC OPEN T#2,STR(F1$,1,8)
 1635 GOTO 1730

-1640 DATA LOAD DC OPEN T#2,STR(F1$,1,8)
 1650 DATA LOAD DC #2,F2$()
 1651 MAT SEARCH F2$(), =Z0$ TO B6$() STEP 8
 1652 IF B6$(1)=HEX(0000) THEN 1660
 1653 GOTO 1750

-1660 F2$(500)=Z0$
 1670 MAT SORT F2$() TO F3$(),F4$()
 1680 MAT MOVE F2$(), F4$() TO F5$()
 1690 MAT COPY F5$() TO F2$()
 1700 GOTO 1730

-1710 DATA SAVE DC OPEN T$#2,(F1$), F1$
 1720 GOTO 1610

-1730 DBACKSPACE #2, BEG
```

```
    : DATA SAVE DC #2,F2$()
 1740 DATA SAVE DC #2,END
-1750 RETURN

 7000 %## PROGRAM ] INPUT UTILITIES    (INPT.SRC) VERSION 2.0  2200 T/VP/MVP
 7106 DEFFN'190(X3,X4,X5,X6)
 7108 DIM X3$(1)1,X0$(80)1
 7110 DIM X8$10,X9(5),X8$(1)1,X$(10)1
 7112 DIM X5$(1)64,X6$(1)2,X4$(1)64
 7114 DIM X8(2,6),X7(3)
 7116 X9(1)=1
 7118 X9(2)=X4-X3+1
 7120 X9(3)=1
 7122 INIT(09)X0$()
 7124 INIT(20)X8$()
 7126 INIT(20)X5$()
 7128 $GIO(010502160388O420,X$())
 7130 $IF ON /0007134
 7132 $GIO(010302000388O420,X$())
-7134 KEYIN X3$(1),7134,7134
 7136 X8=1
 7138 X9=1
 7140 GOSUB '191(X9(1),X9(2))
 7142 GOSUB '101(X9(3)+X3-1,1,0,0)
-7144 MAT COPY X7$()[(X8-1)*X2+X9,1]TO X$()[5,1]
-7146 KEYIN X3$(1),7156,7152
 7148 $GIOCURSOR(7105400642604008121142504008113614541608l146,X$())
 7150 GOTO 7146

-7152 IF X3$(1)=HEX(7E)THEN 7288
 7154 GOSUB '192(X3$(1))
    : GOTO 7144

-7156 IF X3$(1)=HEX(E5)THEN 7276
 7158 IF X3$(1)=HEX(E6)THEN 7288
 7160 IF X3$(1)[]HEX(81)THEN 7164
 7162 INIT(20)X4$(),X5$()
    : GOTO 7144

-7164 IF X3$(1)[=HEX(7A)THEN 7168
 7166 $GIO/005(4007,X8$)
    : GOTO 7144

-7168 IF X3$(1)=HEX(0D)THEN 7188
 7170 IF X3$(1)=HEX(08)THEN 7238
 7172 $GIOPRINT/005(A200,X8$)X3$()[1,1]
 7174 MAT COPY X3$()TO X7$()[(X8-1)*X2+X9,1]
 7176 X9=X9+1
 7178 IF X9[=X6THEN 7182
 7180 IF X8]=X5THEN 7212
-7182 IF X9[]X2-5THEN 7186
 7184 $GIO/005(4007,X8$)
-7186 IF X9[X2+1THEN 7144
-7188 IF X8+1]=X1+1THEN 7212
 7190 IF X8+1]=X5+1THEN 7212
 7192 IF X9(3)+1[=X9(2)THEN 7202
 7194 X9(1)=X9(1)+1
 7196 GOSUB '191(X9(1),X9(2))
 7198 GOSUB '101(X9(3)+X3-1,1,0,0)
 7200 GOTO 7206

-7202 $GIOCRLF/005(400D400A,X8$)
 7204 X9(3)=X9(3)+1
-7206 X8=X8+1
 7208 X9=1
 7210 GOTO 7144

-7212 KEYIN X3$(1),7230,7214
     : GOTO 7212

-7214 IF X3$(1)=HEX(7E)THEN 7288
 7216 IF X3$(1)]HEX(0E)THEN 7236
 7218 IF X3$(1)=HEX(09)THEN 7236
 7220 IF X3$(1)=HEX(0A)THEN 7236
 7222 IF X3$(1)=HEX(02)THEN 7236
 7224 IF X3$(1)=HEX(03)THEN 7236
 7226 IF X3$(1)=HEX(00)THEN 7236
 7228 GOSUB '192(X3$(1))
     : GOTO 7144

-7230 IF X3$(1)=HEX(E6)THEN 7288
 7232 IF X3$(1)=HEX(08)THEN 7238
 7234 IF X3$(1)=HEX(E5)THEN 7276
-7236 $GIO/005(4007,X8$)
     : GOTO 7212

-7238 IF X9[]1THEN 7268
 7240 $GIORUBOUT/005(40204008,X8$)
 7242 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7244 IF X8=1THEN 7144
 7246 X8=X8-1
 7248 X9=X2
 7250 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7252 IF X9(3)[]1THEN 7262
 7254 X9(1)=X9(1)-1
 7256 GOSUB '191(X9(1),X9(2))
 7258 GOSUB '101(X9(3)+X3-1,X2,1,1)
 7260 GOTO 7144

-7262 X9(3)=X9(3)-1
 7264 $GIOUP/005(400CA20040204008,X8$)X0$()[1,X2-1]
 7266 GOTO 7144

-7268 X9=X9-1
 7270 $GIORUBOUT/005(400840204008,X8$)
 7272 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7274 GOTO 7144

-7276 X9=1
 7278 INIT(20)X0$()
 7280 MAT COPY X0$()[1,1]TO X7$()[(X8-1)*X2+1,X2]
 7282 $GIOCLRLINE/005(400DA200400D,X8$)X0$()[1,X2]
 7284 INIT(09)X0$()
 7286 GOTO 7144

-7288 RETURN

 7290 DEFFN'191(X0,X7)
 7292 GOSUB '101(X3,1,0,0)
 7294 FOR X=X0TO X7+X0-1
 7296 IF X=X7+X0-1THEN 7302
 7298 $GIOPRINT/005(A200400D400A,X8$)X7$()[(X-1)*X2+1,X2]
 7300 GOTO 7304

-7302 $GIOPRINT/005(A200,X8$)X7$()[(X-1)*X2+1,X2]
-7304 NEXT X
 7306 RETURN

 7308 DEFFN'192(X3$(1))
 7310 $GIO/005(4005,X8$)
 7312 IF X3$(1)[HEX(01)THEN 7316
 7314 IF X3$(1)[HEX(0F)THEN 7318
-7316 $GIO/005(40074006,X8$)
    : GOTO 7670

-7318 ON VAL(X3$(1))GOTO 7320,7334,7436,7504,7518,7538,7558,7572,7584,7600,761
      4,7628,7642,7656
-7320 X9(1)=(X9(1)+X9(2))-INT((X9(1)+X9(2))/X5)*X5
 7322 IF X9(1)+X9(2)[=X5+1THEN 7326
 7324 X9(1)=X5-X9(2)+1
-7326 GOSUB '191(X9(1),X9(2))
```

```
 7328 X8=X9(1)+X9(3)-1
 7330 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7332 GOTO 7670

-7334 KEYIN X3$(1),7334,7334
 7336 X=0
-7338 KEYIN X3$(1),7346,7340
    : GOTO 7338

-7340 IF X3$(1)=HEX(00)THEN 7344
 7342 IF X3$(1)[]HEX(02)THEN 7364
-7344 X=LEN(X5$(1))
    : GOTO 7356

-7346 IF X3$(1)=HEX(0D)THEN 7356
 7348 IF X3$(1)=HEX(E6)THEN 7364
 7350 X=X+1
 7352 MAT COPY X3$()TO X5$()[X,1]
 7354 IF X[63THEN 7338
-7356 MAT COPY X8$()TO X5$()[X+1,64-X]
 7358 X0=(X8-1)*X2+X9+1
 7360 MAT SEARCHX7$()[X0,(X1*X2)-X0+1],=X5$(1)TO X6$()
 7362 IF X6$(1)[]HEX(0000)THEN 7366
-7364 $GIOBELL/005(4007,X8$)
    : GOTO 7434

-7366 X0=256*VAL(STR(X6$(1),1,1))+VAL(STR(X6$(1),2,1))+X0-1
 7368 X8=INT((X0-1)/X2)+1
 7370 X9=X0-(X8-1)*X2
 7372 IF X8]=X9(1)+X9(2)THEN 7376
 7374 X9(3)=X8-X9(1)+1
    : GOTO 7386

-7376 X9(1)=X8-INT(X9(2)/2)
 7378 IF X9(1)+X9(2)[=X5+1THEN 7382
 7380 X9(1)=X5-X9(2)+1
-7382 X9(3)=X8-X9(1)+1
 7384 GOSUB '191(X9(1),X9(2))
-7386 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7388 IF X3$(1)[]HEX(00)THEN 7434
-7390 KEYIN X3$(1),7390,7390
 7392 X=0
-7394 KEYIN X3$(1),7400,7396
    : GOTO 7394

-7396 IF X3$(1)[]HEX(00)THEN 7364
 7398 X=LEN(X4$(1))
```

```
    : GOTO 7410

-7400 IF X3$(1)=HEX(OD)THEN 7410
 7402 IF X3$(1)=HEX(E6)THEN 7364
 7404 X=X+1
 7406 MAT COPY X3$()TO X4$()[X,1]
 7408 IF X[63THEN 7394
-7410 MAT COPY X8$()TO X4$()[X+1,64-X]
 7412 XO=(X8-1)*X2+X9
 7414 IF XO+LEN(X5$(1))[=X1*X2THEN 7418
 7416 STR(X5$(1),(X1*X2)-XO+1)=" "
-7418 X=LEN(X5$(1))
 7420 MAT COPY X7$()[XO+X,(X1*X2)-(XO+(X-1))]TO X7$()[XO,(X1*X2)-(XO+1)]
 7422 X=LEN(X4$(1))
 7424 X7=(X1*X2)-XO-(X-1)
 7426 MAT COPY -X7$()[XO,X7]TO -X7$()[XO+X,X7]
 7428 MAT COPY X4$()[1,X]TO X7$()[XO,X]
 7430 GOSUB '191(X9(1),X9(2))
 7432 GOSUB '101(X9(3)+X3-1,X9,0,0)
-7434 GOTO 7670

-7436 XO=(X8-1)*X2+X9
 7438 X8(1,1)=X9(1)
    : X8(1,2)=X9(2)
 7440 X8(1,3)=X9(3)
    : X8(1,4)=X8
 7442 X8(1,5)=X9
    : X8(1,6)=XO
-7444 KEYIN X3$(1),7444,7446
    : GOTO 7444

-7446 IF X3$(1)=HEX(03)THEN 7454
 7448 IF X3$(1)=HEX(E6)THEN 7464
 7450 IF X3$(1)=HEX(7E)THEN 7464
 7452 GOSUB '192(X3$(1))
    : GOTO 7444

-7454 XO=(X8-1)*X2+X9
 7456 X8(2,1)=X9(1)
    : X8(2,2)=X9(2)
 7458 X8(2,3)=X9(3)
    : X8(2,4)=X8
 7460 X8(2,5)=X9
    : X8(2,6)=XO
 7462 IF X8(2,6)-X8(1,6)]0THEN 7466
-7464 $GIOBELL/005(4007,X8$)
    : GOTO 7498
```

```
-7466 KEYIN X3$(1),7466,7468
    : GOTO 7466

-7468 IF X3$(1)=HEX(03)THEN 7476
 7470 IF X3$(1)=HEX(E6)THEN 7464
 7472 IF X3$(1)=HEX(7E)THEN 7464
 7474 GOSUB '192(X3$(1))
    : GOTO 7466

-7476 X7(1)=(X8-1)*X2+X9
 7478 X7(2)=X8(2,6)-X8(1,6)+1
 7480 X7=(X1*X2)-X7(1)-(X7(2)-1)
-7482 KEYIN X3$(1),7484,7464
    : GOTO 7482

-7484 IF X3$(1)="R"THEN 7492
 7486 IF X3$(1)="r"THEN 7492
 7488 MAT COPY -X7$()[X7(1),X7]TO -X7$()[X7(1)+X7(2),X7]
 7490 IF X7(1)[=X8(1,6)THEN 7496
-7492 MAT COPY X7$()[X8(1,6),X7(2)]TO X7$()[X7(1),X7(2)]
 7494 GOTO 7498

-7496 MAT COPY X7$()[X8(1,6)+X7(2),X7(2)]TO X7$()[X7(1),X7(2)]
-7498 GOSUB '191(X9(1),X9(2))
 7500 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7502 GOTO 7670

-7504 X9(3)=X9(2)
 7506 X9(1)=X5-X9(2)+1
 7508 X8=X5
 7510 X9=X6
 7512 GOSUB '191(X9(1),X9(2))
 7514 GOSUB '101(X4,X6,0,0)
 7516 GOTO 7670

-7518 IF X8]=X5THEN 7670
 7520 X8=X8+1
 7522 IF X9(3)[X9(2)THEN 7532
 7524 X9(1)=X9(1)+1
 7526 GOSUB '191(X9(1),X9(2))
 7528 GOSUB '101(X4,X9,0,0)
 7530 GOTO 7670

-7532 X9(3)=X9(3)+1
 7534 $GIOLF/005(400A,X8$)
 7536 GOTO 7670
```

```
-7538 IF X8[=1THEN 7670
 7540 X8=X8-1
 7542 IF X9(3)]1THEN 7552
 7544 X9(1)=X9(1)-1
 7546 GOSUB '191(X9(1),X9(2))
 7548 GOSUB '101(X3,X9,0,0)
 7550 GOTO 7670

-7552 X9(3)=X9(3)-1
 7554 $GIOUP/005(400C,X8$)
 7556 GOTO 7670

-7558 X9(3)=1
 7560 X9(1)=1
 7562 X8=1
 7564 X9=1
 7566 GOSUB '191(X9(1),X9(2))
 7568 GOSUB '101(X3,1,0,0)
 7570 GOTO 7670

-7572 X3$(1)=HEX(20)
 7574 X0=(X8-1)*X2+X9
 7576 MAT COPY X3$()[1,1]TO X7$()[X0,(X1*X2)-X0+1]
 7578 GOSUB '191(X9(1),X9(2))
 7580 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7582 GOTO 7670

-7584 X0=(X8-1)*X2+X9
 7586 IF X0[]X1*X2THEN 7592
 7588 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7590 GOTO 7594

-7592 MAT COPY X7$()[X0+1,(X1*X2)-X0]TO X7$()[X0,(X1*X2)-X0+1]
-7594 GOSUB '191(X9(1),X9(2))
 7596 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7598 GOTO 7670

-7600 X0=(X8-1)*X2+X9
 7602 X7=(X1*X2)-X0
 7604 MAT COPY -X7$()[X0,X7]TO -X7$()[X0+1,X7]
 7606 MAT COPY X8$()TO X7$()[(X8-1)*X2+X9,1]
 7608 GOSUB '191(X9(1),X9(2))
 7610 GOSUB '101(X9(3)+X3-1,X9,0,0)
 7612 GOTO 7670

-7614 IF X9+5[=X2THEN 7622
```

```
 7616 X9=1
 7618 $GIOCR/005(400D,X8$)
 7620 GOTO 7670

-7622 X9=X9+5
 7624 $GIORIGHT/005(40094009400940094009,X8$)
 7626 GOTO 7670

-7628 IF X9+1[=X2THEN 7636
 7630 X9=1
 7632 $GIOCR/005(400D,X8$)
 7634 GOTO 7670

-7636 X9=X9+1
 7638 $GIORIGHT/005(4009,X8$)
 7640 GOTO 7670

-7642 IF X9-1]=1THEN 7650
 7644 X9=X2
 7646 $GIOCRTAB/005(400DA200,X8$)X0$()[1,X2-1]
 7648 GOTO 7670

-7650 X9=X9-1
 7652 $GIOLEFT/005(4008,X8$)
 7654 GOTO 7670

-7656 IF X9-5]=1THEN 7664
 7658 X9=X2
 7660 $GIOCRTAB/005(400DA200,X8$)X0$()[1,X2-1]
 7662 GOTO 7670

-7664 X9=X9-5
 7666 $GIOLEFT/005(40084008400840084008,X8$)
 7668 GOTO 7670

-7670 RETURN

 7672 DEFFN'101(Q1,Q2,Q3,Q4)
 7674 DIM Q1$(24)1,Q2$(80)1,Q3$(80)1,Q4$(10)1
 7676 INIT(0A)Q1$()
 7678 INIT(09)Q2$()
 7680 INIT(20)Q3$()
 7682 Q1$(1)=HEX(01)
 7684 Q2$(1)=HEX(0D)
 7686 IF Q3=0THEN 7708
 7688 IF Q4[]0THEN 7692
 7690 Q4=81-Q2
```

```
-7692 FOR Q5=1TO Q3
 7694 IF Q5[]1THEN 7700
 7696 $GIOROW/005(A200,Q4$())Q1$()[1,Q1]
 7698 GOTO 7702

-7700 $GIOROW/005(A200,Q4$())Q1$()[2,1]
-7702 $GIOCOL/005(A200,Q4$())Q2$()[1,Q2]
 7704 $GIOERS/005(A200,Q4$())Q3$()[1,Q4]
 7706 NEXT Q5
-7708 $GIOROW/005(A200,Q4$())Q1$()[1,Q1]
 7710 $GIOCOL/005(A200,Q4$())Q2$()[1,Q2]
 7712 RETURN

 7800 GOSUB ' 239(1)
    : GOSUB ' 239(2)
```

TEXT PRINT PROGRAM (XEXTPRNT)

The text print program will print out a text passage or a whole text file. It prints the text line by line just the way it was keyed in. The program will, however, skip blank lines at the end of each passage in a text file.

The variables and their uses are:

A$(25)80 = Input/Output area for one text passage

A8$3 = Device address where text is stored

F2$(500)8 = Table of text filenames

H$50 = Heading for printout

N9$8 = Filename of text passage or text file

Y1$1 = Yes/No switch to check if table of filenames has been read from disk

Y2$1 = Yes/No switch for end of text passages

Z1$3 = Device address for printer

P1 = Page counter

L4 = Line counter

K = Index to filenames in table of filenames

```
 0005 REM READABILITY SYSTEM TEXT PRINT PROGRAM   (XEXTPRNT)
 0010 REM This program prints the text on the line printer
 0020 DIM A$(25)80,F2$(500)8,A8$3,N9$8,H$50,Z1$3,Y1$1,Y2$1
 0021 SELECT PRINT 005(80)
 0022 PRINT HEX(03)
    : PRINT AT(5,3);"ENTER THE DISK ADDRESS FOR FILE STORAGE"
    : LINPUT -A8$
 0023 SELECT #1 [A8$], #2 [A8$]
 0024 PRINT HEX(03)
    : PRINT AT(5,3);"ENTER THE PASSAGE NAME YOU WISH PRINTED"
    : LINPUT -N9$
 0025 Z1$="215"
    : LINPUT "ENTER OUTPUT DEVICE FOR PRINTING THE TEXT ",Z1$
 0026 PRINT HEX(03)
    : PRINT AT(5,3);"ENTER THE HEADING FOR THE PASSAGE"
    : LINPUT -H$
 0030 SELECT PRINT [Z1$] (80)
 0040 P1=1
    : L4=4
    : Y1$="N"
    : Y2$="N"
 0050 PRINT HEX(0C)
 0060 PRINT TAB(70);"PAGE ";P1
 0062 PRINT TAB(INT((80-LEN(H$)*2)/4));HEX(0E);H$
 0064 PRINT
-0070 IF STR(N9$,5,4)="0000" OR Y1$="Y" THEN GOSUB '110
 0075 IF Y2$="Y" THEN 195
 0080 DATA LOAD DC OPEN T#1,N9$
 0090 DATA LOAD DC #1,A$()
 0091 J=25
 0092 FOR 1=1 TO 25
 0093 IF A$(J) = " " THEN 96
 0094 I=25
 0095 GOTO 97

-0096 J=J-1
-0097 NEXT I
 0100 FOR I=1 TO J
 0105 IF STR(A$(I),1,1)="!" THEN STR(A$(I),1,1)=" "
 0106 IF STR(A$(I),1,1)="[" THEN GOSUB '120
 0110 PRINT A$(I)
 0120 L4=L4+1
 0130 IF L4[56 THEN 180
 0140 P1=P1+1
 0150 PRINT HEX(0C);TAB(70);"PAGE ";P1
 0160 PRINT
 0170 L4=3
```

```
-0180 NEXT I
 0190 IF Y1$="Y" THEN 70
-0195 PRINT HEX(OC)
 0200 SELECT PRINT 005 (80)
 0210 PRINT HEX(03)
 0220 SELECT P3
 0230 FOR I=1 TO 5
 0240 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0250 PRINT HEX(03)
 0260 NEXT I
 0270 SELECT P0
 0280 LOAD DC T#0,"START1"

 0300 DEFFN '110
 0310 IF Y1$="Y" THEN 340
 0320 DATA LOAD DC OPEN T#2,N9$
 0330 DATA LOAD DC #2,F2$()
-0340 K=K+1
 0350 IF STR(F2$(K),1,2)=HEX(FFFF) THEN Y2$="Y"
 0360 N9$=F2$(K)
 0370 Y1$="Y"
 0380 RETURN

 0400 DEFFN '120
 0410 STR(A$(I),1,1)=" "
 0420 P1=P1+1
 0430 PRINT HEX(OC);TAB(70);"PAGE ";P1
 0440 PRINT
 0450 L4=3
 0460 RETURN
```

APPENDIX C

WORD SUBSTITUTION FILE SUBSYSTEM

Program START2 is the menu for the word substitution file subsystem. This program permits the user of the system a means of selecting 1 of 4 programs. If option 1 is selected from the menu by the user, then the program XSUBINPT is loaded into central memory and executed. Program XSUBINPT allows the user to create a new word substitution file. If the user selects option 2, then program XSUBEDIT is loaded and executed. Program XSUBEDIT permits the user to edit an existing word substitution list. Upon selection of option 3, the program XSUBPRNT is loaded and executed. This program prints out the contents of a word substitution file. Option 4 provides a means for the user to load and execute the program XPNDROOT. This program allows the user to enter a root word and the program automatically produces and displays a list of standard endings of the root word. Selection of option 5 returns the system to the master menu.

```
 0010 %## WORD SUBSTITUTION FILE SUBSYSTEM MENU PROGRAM  START2
-0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
    : PRINTUSING 580
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633,634
 0390 IF O=99 THEN 638
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %             * * *  WORD SUBSTITUTION SUBSYSTEM MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1     -  LOAD WORD SUBSTITUTION FILE INPUT PROGRAM
-0565 %     2     -  LOAD WORD SUBSTITUTION FILE EDIT PROGRAM
-0575 %     3     -  LOAD WORD SUBSTITUTION FILE PRINT PROGRAM
-0580 %     4     -  LOAD ADD EXPANDED ROOT WORDS PROGRAM
-0630 %    99     -  ** Return to Master Menu  **
-0631 LOAD DC T#0, "XSUBINPT"

-0632 LOAD DC T#0, "XSUBEDIT"

-0633 LOAD DC T#0, "XSUBPRNT"

-0634 LOAD DC T#0,"XPNDROOT"

-0638 LOAD DC T#0, "START" BEG 310
```

WORD SUBSTITUTION FILE INPUT PROGRAM (XSUBINPT)

The word substitution file input program allows the user to create a new word substitution file. The user can also delete words or change substitutes of words he has already entered on the new file. The word substitution file is made up of blocks of 200 records of 60 characters each. The records are broken down into three parts. Characters 1-25 are the original word, characters 26-45 are the first substitute and characters 46-60 are the second substitute. Either the first or second substitute or both may be blank.

This program creates a block pointer file for each word substitution file. The block pointer file consists of a table of 100 elements of 3 characters each. These elements are the first three characters of the first word in each block of the word substitution file. This program also requires a work area of 4000 sectors called "TEMPWORK". This work area is used to merge new blocks of words with the words that have already been entered.

After each block of 200 words has been entered, the program sorts the current block of words and then merges it with any words that have already been entered. At this time the program also updates the block pointer file.

The variables used in this program are as follows:

A$(200)60 = One input/output block of words with their substitutes
A1$3 = Device address of the word substitution and block pointer files
A2$3 = Device address of work file "TEMPWORK"
A3$8 = Filename of the word substitution file
A4$8 = Filename of the block pointer file
B$(200)60 = Current input block of words with their substitutes
B1$(2)2 = Locator array for MAT SEARCH statements
C$(200)60 = Output array for merge of word substitution file and current block of words

F3$(200)2 = Work area for MAT SORT of B$()

F4$(200)2 = Output locator array for MAT SORT of B$()

P$(100)3 = Block pointer table

W1$25 = Input word

W2$20 = First substitute

W3$15 = Second substitute

Z1$1 = Yes/No switch to check if word has already been entered

Z2$1 = Option number to delete, change substitutes, or leave word alone

Z3$1 = Yes/No switch for end of file during merge of word substitution file and current block of words

```
 0010 % ## PROGRAM TO INPUT WORD SUBSTITUTION FILES  (XSUBINPT)
 0020 DIM A$(200)60,A1$3,A2$3,A3$8,B$(200)60,W1$25,W2$20,W3$15,P$(100)3
 0030 DIM Z1$1,Z2$1,B1$(2)2,F3$(200)2,F4$(200)2,C$(200)60,Z3$1,A4$8
 0040 SELECT PRINT 005(80)
 0050 PRINT HEX(03)
 0060 LINPUT "ENTER THE DEVICE OF THE WORD SUBSTITUTION FILE",-A1$
 0070 LINPUT "ENTER THE DEVICE OF THE WORK FILE",-A2$
 0080 PRINT HEX(03)
 0090 A4$="BLKPOINT"
 0100 LINPUT "ENTER THE FILENAME OF THE BLOCK POINTER FILE",-A4$
-0110 LINPUT "ENTER THE FILENAME OF THE WORD SUBSTITUTION FILE",-A3$
 0120 PRINT HEX(03)
 0130 SELECT #1 [A1$], #2 [A2$], #3 [A1$]
 0140 LIMITS T#1,A3$,U1,U2,U3,S1
 0150 IF S1[]2 THEN 170
 0160 STOP "WORD SUBSTITUTION FILE IS AN ACTIVE FILE"
-0170 IF S1=0 THEN 200
 0172 IF S1=-2 THEN 180
 0173 PRINT "INVALID FILENAME.....RE-ENTER";HEX(07)
 0174 GOTO 110

-0180 DATA SAVE DC OPEN T$#1,(A3$),A3$
 0190 GOTO 210

-0200 DATA SAVE DC OPEN T$#1,(3000),A3$
-0210 DATA LOAD DC OPEN T#1,A3$
 0220 LIMITS T#2,"TEMPWORK",L1,L2,L3,S
 0230 IF S=2 THEN 320
 0240 IF S=0 THEN 290
 0250 IF S=-2 THEN 310
 0260 PRINT "TEMPWORK IS CATALOGED AS A PROGRAM FILE"
 0270 PRINT "THIS PROGRAM REQUIRES A WORK AREA CALLED TEMPWORK"
 0280 STOP "A PROGRAMMER MUST CHANGE THIS PROGRAM OR THE DISK CATALOG"
-0290 DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"
 0300 GOTO 320

-0310 DATA SAVE DC OPEN T$#2,("TEMPWORK"),"TEMPWORK"
-0320 DATA LOAD DC OPEN T#2,"TEMPWORK"
 0330 LIMITS T#3,A4$,U4,U5,U6,S2
 0340 IF S2[]2 THEN 360
 0350 STOP "BLOCK POINTER FILE IS AN ACTIVE FILE"
-0360 IF S2=0 THEN 390
 0370 DATA SAVE DC OPEN T$#3,(A4$),A4$
 0380 GOTO 400

-0390 DATA SAVE DC OPEN T$#3,(4),A4$
-0400 DATA LOAD DC OPEN T#3,A4$
```

```
 0410 INIT(FF) B$()
 0420 DATA SAVE DC #1,A$()
 0430 DATA SAVE DC #1,END
 0440 LIMITS T#1,A3$,U1,U2,U3,S1
 0450 P$(1)="A  "
 0460 P$(2)="ZZZ"
-0470 PRINT HEX(03)
    : INIT(20) W1$,W2$,W3$
    : Z2$=" "
-0480 LINPUT "ENTER WORD OR EOL",-W1$
 0490 IF W1$[]" " THEN 520
 0500 PRINT HEX(07)
 0510 GOTO 480

-0520 IF W1$="EOL" THEN 1010
 0530 GOSUB '110
 0540 IF Z1$="N" THEN 730
 0550 PRINT "WORD IS ALREADY ON FILE"
-0560 LINPUT "ENTER 1) TO DELETE WORD 2) TO CHANGE SUBSTITUTES 3) TO LEAVE WOR
      D UNCHANGED",-Z2$
 0570 IF Z2$="2" THEN 660
 0580 IF Z2$="1" THEN 630
 0590 IF Z2$="3" THEN 470
 0600 Z2$=" "
 0610 PRINT HEX(07)
 0620 GOTO 560

-0630 INIT(FF) A$(Q)
 0640 DATA SAVE DC #1,A$()
 0650 GOTO 470

-0660 PRINT HEX(03)
 0670 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0680 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0690 STR(A$(Q),26,20)=W2$
 0700 STR(A$(Q),46,15)=W3$
 0710 DATA SAVE DC #1,A$()
 0720 GOTO 470

-0730 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0740 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0750 MAT SEARCH B$(), =STR(W1$,1,25) TO B1$() STEP 60
 0760 IF B1$(1)[]HEX(0000) THEN 830
 0770 V=V+1
 0780 IF V=201 THEN GOSUB '120
 0790 STR(B$(V),1,25)=W1$
 0800 STR(B$(V),26,20)=W2$
 0810 STR(B$(V),46,15)=W3$
 0820 GOTO 470

-0830 PRINT "WORD IS ALREADY ON FILE"
-0840 LINPUT "ENTER 1) TO DELETE WORD 2) TO CHANGE SUBSTITUTES 3) TO LEAVE WOR
      D UNCHANGED",-Z2$
 0850 IF Z2$="3" THEN 470
 0860 IF Z2$="2" THEN 930
 0870 IF Z2$="1" THEN 900
 0880 Z2$=" "
 0890 GOTO 840

-0900 V1=INT(VAL(B1$(1),2)/60)+1
 0910 INIT(FF) B$(V1)
 0920 GOTO 470

-0930 V1=INT(VAL(B1$(1),2)/60)+1
 0940 PRINT HEX(03)
 0950 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0960 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0970 STR(B$(V1),1,25)=W1$
 0980 STR(B$(V1),26,20)=W2$
 0990 STR(B$(V1),46,15)=W3$
 1000 GOTO 470

-1010 IF V=0 THEN 1040
 1020 IF V=1 AND STR(B$(1),1,2)=HEX(FFFF) THEN 1040
 1030 GOSUB '120
-1040 PRINT HEX(03)
 1050 SELECT P3
 1060 FOR I=1 TO 5
 1070 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 1080 NEXT I
 1090 SELECT P0
 1100 LOAD DC T#0,"START2"

 1110 DEFFN '110
 1120 Z1$="N"
 1130 MAT SEARCH P$(), ]STR(W1$,1,3) TO B1$() STEP 3
 1140 P6=INT(VAL(B1$(1),2)/3)
-1150 DATA LOAD DA T#1, (U1+(P6-1)*50) A$()
 1160 MAT SEARCH A$(), =STR(W1$,1,25) TO B1$() STEP 60
 1170 IF B1$(1)=HEX(0000) THEN 1240
 1180 Z1$="Y"
 1190 Q=INT(VAL(B1$(1),2)/60)+1
 1200 W2$=STR(A$(Q),26,20)
 1210 W3$=STR(A$(Q),46,15)
 1220 DBACKSPACE #1,BEG
 1225 DSKIP #1,((P6-1)*50)S
 1230 GOTO 1280

-1240 IF STR(W1$,1,3)[] STR(A$(1),1,3) THEN 1280
 1250 IF P6=1 THEN 1280
 1260 P6=P6-1
 1270 GOTO 1150

-1280 RETURN

 1290 DEFFN '120
 1300 Z3$="N"
 1310 MAT SORT B$() TO F3$(),F4$()
 1320 MAT MOVE B$(),F4$() TO A$()
 1330 MAT COPY A$() TO B$()
 1340 DBACKSPACE #1,BEG
 1350 PRINT HEX(03)
 1360 PRINT AT(10,10);"ONE MOMENT PLEASE WHILE I UPDATE ";A3$
 1370 I=1
 1380 J=1
 1390 K=1
 1400 DATA LOAD DC #1,A$()
-1410 IF A$(K)=" " THEN GOSUB '150
 1420 IF Z3$="Y" AND STR(B$(J),1,2)=HEX(FFFF) THEN 1770
 1430 IF STR(A$(K),1,25)[STR(B$(J),1,25) THEN 1570
 1440 IF STR(A$(K),1,2)[]HEX(FFFF) THEN 1490
 1450 IF Z3$="Y" THEN 1490
 1460 K=K+1
 1470 IF K=201 THEN GOSUB '150
 1480 GOTO 1410

-1490 C$(I)=B$(J)
 1500 I=I+1
 1510 IF I=201 THEN GOSUB '160
 1520 J=J+1
 1530 IF J[]201 THEN 1410
 1540 J=1
 1550 INIT(FF) B$()
 1560 GOTO 1410

-1570 C$(I)=A$(K)
 1580 I=I+1
 1590 IF I=201 THEN GOSUB '160
 1600 K=K+1
 1610 IF K=201 THEN GOSUB '150
 1620 GOTO 1410
 1630 DEFFN '150
 1640 DATA LOAD DC #1,A$()
 1650 IF END THEN 1680
 1660 K=1
 1670 GOTO 1710

-1680 Z3$="Y"
 1690 INIT(FF) A$()
 1700 K=1
-1710 RETURN

 1720 DEFFN '160
 1730 DATA SAVE DC #2,C$()
 1740 INIT(20)C$()
 1750 I=1
 1760 RETURN

-1770 IF I=1 THEN 1790
 1780 DATA SAVE DC #2,C$()
-1790 DATA SAVE DC #2,END
 1800 DBACKSPACE #1,BEG
 1810 DBACKSPACE #2,BEG
 1820 DBACKSPACE #3,BEG
 1830 INIT(20) P$()
 1840 I=1
-1850 DATA LOAD DC #2,A$()
 1860 IF END THEN 1940
 1870 IF I[]1 THEN 1900
 1880 P$(1)="A  "
 1890 GOTO 1910

-1900 P$(I)=STR(A$(1),1,3)
-1910 DATA SAVE DC #1,A$()
 1920 I=I+1
 1930 GOTO 1850

-1940 DATA SAVE DC #1,END
 1950 P$(I)="ZZZ"
 1960 DATA SAVE DC #3,P$()
 1970 DATA SAVE DC #3,END
 1980 DBACKSPACE #1,BEG
 1990 DBACKSPACE #2,BEG
 2000 DBACKSPACE #3,BEG
 2010 V=1
 2020 INIT(FF) B$()
 2030 RETURN
```

WORD SUBSTITUTION FILE EDIT PROGRAM (XSUBEDIT)

The word substitution file edit program allows the user to edit a word substitution file that is already on disk. The user can add or delete words, or change substitutes for words already on file. This program works just like the word substitution input program except that the word substitution file and the block pointer file must already be on disk. The variables for this program are the same as for the word substitution input program.

```
 0005 % ## PROGRAM TO EDIT WORD SUBSTITUTION FILES  (XSUBEDIT)
 0010 DIM A$(200)60,A1$3,A2$3,A3$8,B$(200)60,W1$25,W2$20,W3$15,P$(100)3
 0020 DIM Z1$1,Z2$1,B1$(2)2,F3$(200)2,F4$(200)2,C$(200)60,Z3$1,A4$8
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 LINPUT "ENTER THE DEVICE OF THE WORD SUBSTITUTION FILE",-A1$
 0130 LINPUT "ENTER THE DEVICE OF THE WORK FILE",-A2$
 0140 PRINT HEX(03)
 0145 A4$="BLKPOINT"
 0150 LINPUT "ENTER THE FILENAME OF THE BLOCK POINTER FILE",-A4$
-0155 LINPUT "ENTER THE FILENAME OF THE WORD SUBSTITUTION FILE",-A3$
 0160 PRINT HEX(03)
 0170 SELECT #1 [A1$], #2 [A2$], #3 [A1$]
 0181 LIMITS T#1,A3$,U1,U2,U3,S1
 0182 IF S1=2 THEN 188
 0185 PRINT "INVALID FILENAME....RE-ENTER";HEX(07)
 0186 GOTO 155

-0188 DATA LOAD DC OPEN T#1,A3$
 0190 LIMITS T#2,"TEMPWORK",L1,L2,L3,S
 0200 IF S=2 THEN 300
 0210 IF S=0 THEN 260
 0220 IF S=-2 THEN 280
 0230 PRINT "TEMPWORK IS CATALOGED AS A PROGRAM FILE"
 0240 PRINT "THIS PROGRAM REQUIRES A WORK AREA CALLED TEMPWORK"
 0250 STOP "A PROGRAMMER MUST CHANGE THIS PROGRAM OR THE DISK CATALOG"
-0260 DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"
 0270 GOTO 300

-0280 DATA SAVE DC OPEN T$#2,("TEMPWORK"),"TEMPWORK"
-0300 DATA LOAD DC OPEN T#2,"TEMPWORK"
 0310 DATA LOAD DC OPEN T#3,A4$
 0320 INIT(FF) B$()
 0330 DATA LOAD DC #3,P$()
-0335 PRINT HEX(03)
    : INIT(20) W1$,W2$,W3$
    : Z2$=" "
-0340 LINPUT "ENTER WORD OR EOL",-W1$
 0342 IF W1$[]" " THEN 350
 0344 PRINT HEX(07)
 0346 GOTO 340

-0350 IF W1$="EOL" THEN 850
 0360 GOSUB '110
 0370 IF Z1$="N" THEN 550
 0380 PRINT "WORD IS ALREADY ON FILE"
-0390 LINPUT "ENTER 1) TO DELETE WORD 2) TO CHANGE SUBSTITUTES 3) TO LEAVE WOR
```

```
     D UNCHANGED",-Z2$
 0400 IF Z2$="2" THEN 480
 0410 IF Z2$="1" THEN 450
 0415 IF Z2$="3" THEN 335
 0420 Z2$=" "
 0430 PRINT HEX(07)
 0440 GOTO 390

-0450 INIT(FF) A$(Q)
 0460 DATA SAVE DC #1,A$()
 0470 GOTO 335

-0480 PRINT HEX(03)
 0490 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0500 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0510 STR(A$(Q),26,20)=W2$
 0520 STR(A$(Q),46,15)=W3$
 0530 DATA SAVE DC #1,A$()
 0540 GOTO 335

-0550 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0560 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0570 MAT SEARCH B$(), =STR(W1$,1,25) TO B1$() STEP 60
 0580 IF B1$(1)[]HEX(0000) THEN 650
 0590 V=V+1
 0600 IF V=201 THEN GOSUB '120
 0610 STR(B$(V),1,25)=W1$
 0620 STR(B$(V),26,20)=W2$
 0630 STR(B$(V),46,15)=W3$
 0640 GOTO 335

-0650 PRINT "WORD IS ALREADY ON FILE"
-0660 LINPUT "ENTER 1) TO DELETE WORD 2) TO CHANGE SUBSTITUTES 3) TO LEAVE WOR
     D UNCHANGED",-Z2$
 0670 IF Z2$="3" THEN 335
 0680 IF Z2$="2" THEN 750
 0690 IF Z2$="1" THEN 720
 0700 Z2$=" "
 0710 GOTO 660

-0720 V1=INT(VAL(B1$(1),2)/60)+1
 0730 INIT(FF) B$(V1)
 0740 GOTO 335

-0750 V1=INT(VAL(B1$(1),2)/60)+1
 0760 PRINT HEX(03)
 0770 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-W2$
 0780 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-W3$
 0790 STR(B$(V1),1,25)=W1$
 0800 STR(B$(V1),26,20)=W2$
 0810 STR(B$(V1),46,15)=W3$
 0820 GOTO 335

-0850 IF V=0 THEN 880
 0860 IF V=1 AND STR(B$(1),1,2)=HEX(FFFF) THEN 880
 0870 GOSUB '120
-0880 PRINT HEX(03)
 0890 SELECT P3
 0900 FOR I=1 TO 5
 0910 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0920 NEXT I
 0930 SELECT P0
 0940 LOAD DC T#0,"START2"

 1000 DEFFN '110
 1010 Z1$="N"
 1020 MAT SEARCH P$(), ]STR(W1$,1,3) TO B1$() STEP 3
 1030 P6=INT(VAL(B1$(1),2)/3)
-1040 DATA LOAD DA T#1, (U1+(P6-1)*50) A$()
 1050 MAT SEARCH A$(), =STR(W1$,1,25) TO B1$() STEP 60
 1060 IF B1$(1)=HEX(0000) THEN 1120
 1070 Z1$="Y"
 1080 Q=INT(VAL(B1$(1),2)/60)+1
 1090 W2$=STR(A$(Q),26,20)
 1100 W3$=STR(A$(Q),46,15)
 1105 DBACKSPACE #1,BEG
 1106 DSKIP #1,((P6-1)*50)S
 1110 GOTO 1150

-1120 IF STR(W1$,1,3)[] STR(A$(1),1,3) THEN 1150
 1125 IF P6=1 THEN 1150
 1130 P6=P6-1
 1140 GOTO 1040

-1150 RETURN

 1200 DEFFN '120
 1205 Z3$="N"
 1210 MAT SORT B$() TO F3$(),F4$()
 1220 MAT MOVE B$(),F4$() TO A$()
 1230 MAT COPY A$() TO B$()
 1240 DBACKSPACE #1,BEG
 1250 PRINT HEX(03)
 1260 PRINT AT(10,10);"ONE MOMENT PLEASE WHILE I UPDATE ";A3$
 1270 I=1
 1280 J=1
 1290 K=1
 1300 DATA LOAD DC #1,A$()
-1310 IF A$(K)=" " THEN GOSUB '150
 1315 IF Z3$="Y" AND STR(B$(J),1,2)=HEX(FFFF) THEN 1580
 1320 IF STR(A$(K),1,25)[STR(B$(J),1,25) THEN 1380
 1322 IF STR(A$(K),1,2)[]HEX(FFFF) THEN 1330
 1323 IF Z3$="Y" THEN 1330
 1324 K=K+1
 1325 IF K=201 THEN GOSUB '150
 1326 GOTO 1310

-1330 C$(I)=B$(J)
 1340 I=I+1
 1350 IF I=201 THEN GOSUB '160
 1360 J=J+1
 1365 IF J[]201 THEN 1310
 1370 J=1
 1372 INIT(FF) B$()
 1375 GOTO 1310

-1380 C$(I)=A$(K)
 1390 I=I+1
 1400 IF I=201 THEN GOSUB '160
 1410 K=K+1
 1420 IF K=201 THEN GOSUB '150
 1430 GOTO 1310

 1440 DEFFN '150
 1450 DATA LOAD DC #1,A$()
 1460 IF END THEN 1490
 1470 K=1
 1480 GOTO 1520

-1490 Z3$="Y"
 1500 INIT(FF) A$()
 1510 K=1
-1520 RETURN

 1530 DEFFN '160
 1540 DATA SAVE DC #2,C$()
 1550 INIT(20)C$()
 1560 I=1
 1570 RETURN

-1580 IF I=1 THEN 1600
```

```
 1590 DATA SAVE DC #2,C$()
-1600 DATA SAVE DC #2,END
 1610 DBACKSPACE #1,BEG
 1620 DBACKSPACE #2,BEG
 1630 DBACKSPACE #3,BEG
 1640 INIT(20) P$()
 1650 I=1
-1660 DATA LOAD DC #2,A$()
 1670 IF END THEN 1750
 1680 IF I[]1 THEN 1710
 1690 P$(1)="A  "
 1700 GOTO 1720

-1710 P$(I)=STR(A$(1),1,3)
-1720 DATA SAVE DC #1,A$()
 1730 I=I+1
 1740 GOTO 1660

-1750 DATA SAVE DC #1,END
 1755 P$(I)="ZZZ"
 1760 DATA SAVE DC #3,P$()
 1770 DATA SAVE DC #3,END
 1780 DBACKSPACE #1,BEG
 1790 DBACKSPACE #2,BEG
 1800 DBACKSPACE #3,BEG
 1810 V=1
 1820 INIT(FF) B$()
 1830 RETURN
```

WORD SUBSTITUTION FILE PRINT PROGRAM (XSUBPRNT)

The word substitution file print program allows the user to print out the contents of a word substitution file. It is a simple disk-to-print program with headings. The variables are as follows:

A$(200)60 = Input block of words with their substitutes

A1$3 = Device address where word substitution file is stored

A2$8 = Filename of word substitution file

A3$3 = Device address of printer

A4$40 = Title for printout

W1$25 = Output area for a word from the file

W2$20 = Output area for first substitute

W3$15 = Output area for second substitute

Z1$1 = Yes/No switch for title

```
 0010 %## PROGRAM ]  WORD SUBSTITUTION FILE LIST   (XSUBPRNT)
 0020 DIM A$(200)60,A1$3,A2$8,A3$3,Z1$1,A4$40
 0030 DIM W1$25,W2$20,W3$15
 0100 L1=99
 0110 SELECT PRINT 005(80)
 0120 PRINT HEX(03)
 0130 LINPUT "ENTER DEVICE FOR FILE STORAGE",-A1$
 0135 SELECT #1 [A1$]
-0140 LINPUT "ENTER FILENAME OF WORD SUBSTITUTION FILE",-A2$
 0144 LIMITS T#1,STR(A2$,1,8),L1,L2,L3,S
 0145 IF S=2 THEN 150
 0146 PRINT "INVALID FILENAME...."
 0147 A2$=" "
 0148 GOTO 140

-0150 LINPUT "ENTER DEVICE FOR PRINTING LIST",-A3$
 0160 PRINT HEX(03)
 0162 LINPUT "DO YOU WISH TO TITLE THIS LIST? (Y/N)",-Z1$
 0163 IF Z1$="N" THEN 169
 0164 LINPUT "ENTER TITLE ",-A4$
-0169 PRINT HEX(03)
 0170 PRINT AT(10,10);"PRINTING FILE NOW...."
 0175 SELECT PRINT [A3$] (132)
 0190 DATA LOAD DC OPEN T#1,A2$
-0200 DATA LOAD DC #1,A$()
 0205 IF END THEN 420
 0210 FOR I=1 TO 200
 0220 IF L1]55 THEN 290
 0225 IF A$(I)=" " THEN 380
-0230 W1$=STR(A$(I),1,25)
 0240 W2$=STR(A$(I),26,20)
 0250 W3$=STR(A$(I),46,15)
 0260 PRINTUSING 390,W1$,W2$,W3$
 0270 L1=L1+1
 0280 GOTO 380

-0290 PRINT HEX(0C)
 0300 L1=5
 0310 C8=INT((80-LEN(A4$)*2)/4)
 0320 PRINT TAB(C8);HEX(0E);A4$
 0330 PRINT
 0340 PRINTUSING 400
 0350 PRINTUSING 410
 0360 PRINTUSING 400
 0370 PRINT
 0375 GOTO 230
```

```
-0380 NEXT I
 0385 GOTO 200

-0390  %   #########################     ####################     ############
      ###
-0400   %=====================================================================
       ======
-0410  %        WORD                     FIRST SUBSTITUTE    SECOND SUBSTITUT
      E
-0420 SELECT PRINT 005(80)
 0430 PRINT HEX(03)
 0440 SELECT P3
 0450 FOR I=1 TO 5
 0460 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0470 NEXT I
 0480 SELECT P0
 0490 LOAD DC T#0,"START2"
```

ADD EXPANDED WORDS TO WORD SUBSTITUTION FILE (XPNDROOT)

This program allows the user to input common words and have them added to the specified word substitution file. After the user enters a root word, a list of that word with all the standard endings is displayed on the console. The user can then erase the bad words or make changes to the spelling of words. For example, "standed" could be changed to "stood". If the user wants the underlined word added to the word substitution file then he should press return. If he wants to change the underlined word he should key in the word the way he wants it and then press return. To delete the underlined word he should press erase and then return. Suggested substitutes can not be added with this program. If the user wishes to add a substitute then he should go back and use the word substitution file edit program after he is finished with this program. The variables in this program are as follows:

A$(200)60 = One input/output block of words

A1$3 = Device address of the word substitution file

A2$3 = Device address of the work file

A3$8 = Filename of the word substitution file

A4$8 = Filename of the block pointer file

B$(200)60 = Input table of root and expanded words

B1$(2)2 = Locator variable for MAT SEARCH

C$20 = Table of consonants

C$(200)60 = Output block of words for merge

F3$(200)2 = Work area for MAT SORT

F4$(200)2 = Locator array for MAT SORT and MAT MOVE

P$(100)3 = Block pointer table

W1$25 = Root word

W2$(12)25 = Table of expanded words

Z1$1 = Yes/No switch to check if list of expanded words is correct

Z3$1 = Yes/No switch for end of file during merge

```
 0010 %### PROGRAM ]  EXPAND ROOT WORDS  (XPNDROOT)
 0015 DIM A$(200)60,A1$3,A2$3,A3$8,B$(200)60,P$(100)3,A4$8,B1$(2)2
 0016 DIM F3$(200)2,F4$(200)2,C$(200)60,Z3$1
 0020 DIM W1$25,W2$(12)25,C$20,Z1$1
 0030 C$="BCDFGHJKLMNPQRSTVWXZ"
 0040 SELECT PRINT 005(80)
 0050 PRINT HEX(03)
 0060 LINPUT "ENTER THE DEVICE OF THE WORD SUBSTITUTION FILE",-A1$
 0070 LINPUT "ENTER THE DEVICE OF THE WORK FILE",-A2$
 0080 PRINT HEX(03)
 0090 A4$="BLKPOINT"
 0100 LINPUT "ENTER THE FILENAME OF THE BLOCK POINTER FILE",-A4$
-0110 LINPUT "ENTER THE FILENAME OF THE WORD SUBSTITUTION FILE",-A3$
 0120 PRINT HEX(03)
 0130 SELECT #1 [A1$], #2 [A2$], #3 [A1$]
 0135 DATA LOAD DC OPEN T#1,A3$
 0140 LIMITS T#1,A3$,U1,U2,U3,S1
 0150 IF S1=2 THEN 180
 0160 PRINT "INVALID FILENAME....RE-ENTER";HEX(07)
 0170 GOTO 110

-0180 LIMITS T#2,"TEMPWORK",L1,L2,L3,S
 0190 IF S=2 THEN 300
 0200 IF S=0 THEN 250
 0210 IF S=-2 THEN 270
 0220 PRINT "TEMPWORK IS CATALOGED AS A PROGRAM FILE"
 0230 PRINT "THIS PROGRAM REQUIRES A WORK AREA CALLED TEMPWORK"
 0240 STOP "A PROGRAMMER MUST CHANGE THIS PROGRAM OR THE DISK CATALOG"
-0250 DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"
 0260 GOTO 300

-0270 DATA SAVE DC OPEN T$#2,("TEMPWORK"),"TEMPWORK"
-0300 DATA LOAD DC OPEN T#2,"TEMPWORK"
 0310 DATA LOAD DC OPEN T#3,A4$
 0320 INIT(FF) B$()
-0340 W1$=" "
 0350 PRINT HEX(03)
 0360 LINPUT "ENTER ROOT WORD OR EOL",-W1$
 0370 IF W1$="EOL" THEN 1040
 0380 INIT(20) W2$()
 0390 FOR I=1 TO 12
 0400 W2$(I)=W1$
 0410 NEXT I
 0420 W2$(1)=W2$(1) & "S"
 0430 IF STR(W2$(2),LEN(W2$(2)),1) = "Y" THEN STR(W2$(2),LEN(W2$(2)),1)="I"
 0440 W2$(2)=W2$(2) & "ES"
 0450 IF STR(W2$(3),LEN(W2$(3)),1) = "Y" THEN STR(W2$(3),LEN(W2$(3)),1)="I"
 0460 IF STR(W2$(3),LEN(W2$(3)),1) = "E" THEN W2$(3)= STR(W2$(3),1,LEN(W2$(3))
      -1)
 0470 W2$(3)=W2$(3) & "ED"
 0480 W2$(4)=W2$(4) & "ING"
 0490 IF POS(C$=STR(W2$(5),LEN(W2$(5)),1))=0 THEN 510
 0500 W2$(5)=W2$(5) & STR(W2$(5),LEN(W2$(5)),1)
-0510 IF STR(W2$(5),LEN(W2$(5)),1)="E" THEN W2$(5)=STR(W2$(5),1,LEN(W2$(5))-1)

 0520 W2$(5)=W2$(5) & "ING"
 0530 W2$(6)=W2$(6) & "Y"
 0540 W2$(7)=W2$(7) & "LY"
 0550 W2$(8)=W2$(8) & "ILY"
 0560 W2$(9)=W2$(9) & "ER"
 0570 W2$(10)= STR(W2$(5),1,LEN(W2$(5))-3) & "ER"
 0580 W2$(11)= W2$(11) & "EST"
 0590 W2$(12)= STR(W2$(5),1,LEN(W2$(5))-3) & "EST"
 0600 PRINT AT(4,11);W1$
 0610 FOR I=1 TO 12
 0620 PRINT AT(I+4,11);W2$(I)
 0630 NEXT I
-0640 PRINT AT(18,1,);"EDIT THE UNDERLINED WORD AND THEN PRESS RETURN"
 0650 PRINT AT(20,1);"PRESS ERASE THEN RETURN TO DELETE THE UNDERLINED WORD"
 0660 PRINT AT(4,10);
 0670 LINPUT -W1$
 0680 FOR I=1 TO 12
 0690 PRINT AT(I+4,10);
 0700 LINPUT -W2$(I)
 0710 NEXT I
 0720 Z1$="Y"
-0730 PRINT AT(18,1,);"ENTER Y IF LIST IS CORRECT"
 0740 PRINT AT(20,1);
 0750 LINPUT "ENTER N TO CHANGE THE LIST",-Z1$
 0760 IF Z1$="N" THEN 640
 0770 IF Z1$="Y" THEN 900
 0780 PRINT HEX(07)
 0790 GOTO 730

-0900 MAT SEARCH B$(), =STR(W1$,1,25) TO B1$() STEP 60
 0910 IF B1$(1)[]HEX(0000) THEN 930
 0920 B$(200)=W1$
-0930 FOR I=1 TO 12
 0940 IF W2$(I)=" " THEN 980
 0950 MAT SEARCH B$(), =STR(W2$(I),1,25) TO B1$() STEP 60
 0960 IF B1$(1)[]HEX(0000) THEN 980
 0970 B$(200-I)=W2$(I)
-0980 NEXT I
 0990 MAT SORT B$() TO F3$(), F4$()
 1000 MAT MOVE B$(),F4$() TO A$()
 1010 MAT COPY A$() TO B$()
 1020 IF STR(B$(180),1,2)[]HEX(FFFF) THEN GOSUB '110
 1030 GOTO 340

-1040 IF STR(B$(1),1,2)[]HEX(FFFF) THEN GOSUB '110
 1050 PRINT HEX(03)
 1060 SELECT P3
 1070 FOR I=1 TO 5
 1080 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 1090 NEXT I
 1100 SELECT P0
 1110 LOAD DC T#0,"START2"

 1120 DEFFN '110
 1125 Z3$="N"
 1130 DBACKSPACE #1,BEG
 1140 PRINT HEX(03)
 1150 PRINT AT(10,10);"ONE MOMENT PLEASE WHILE I UPDATE ";A3$
 1160 I=1
 1170 J=1
 1180 K=1
 1190 DATA LOAD DC #1,A$()
-1200 IF A$(K)=" " THEN GOSUB '150
 1210 IF Z3$="Y" AND STR(B$(J),1,2)=HEX(FFFF) THEN 1670
 1220 IF STR(A$(K),1,25)[STR(B$(J),1,25) THEN 1470
 1230 IF STR(A$(K),1,25)=STR(B$(J),1,25) AND STR(A$(K),1,2)[]HEX(FFFF) THEN 13
      70
 1240 IF STR(A$(K),1,2)[]HEX(FFFF) THEN 1290
 1250 IF Z3$="Y" THEN 1290
 1260 K=K+1
 1270 IF K=201 THEN GOSUB '150
 1280 GOTO 1200

-1290 C$(I)=B$(J)
 1300 I=I+1
 1310 IF I=201 THEN GOSUB '160
 1320 J=J+1
 1330 IF J[]201 THEN 1200
 1340 J=1
 1350 INIT(FF) B$()
 1360 GOTO 1200

-1370 C$(I)=A$(K)
 1380 I=I+1
 1390 IF I=201 THEN GOSUB '160
 1400 K=K+1
 1410 IF K=201 THEN GOSUB '150
 1420 J=J+1
 1430 IF J[]201 THEN 1200
 1440 J=1
 1450 INIT(FF) B$()
 1460 GOTO 1200

-1470 C$(I)=A$(K)
 1480 I=I+1
 1490 IF I=201 THEN GOSUB '160
 1500 K=K+1
 1510 IF K=201 THEN GOSUB '150
 1520 GOTO 1200

 1530 DEFFN '150
 1540 DATA LOAD DC #1,A$()
 1550 IF END THEN 1580
 1570 GOTO 1600

-1580 Z3$="Y"
 1590 INIT(FF) A$()
-1600 K=1
 1610 RETURN

 1620 DEFFN '160
 1630 DATA SAVE DC #2,C$()
 1640 INIT(20) C$()
 1650 I=1
 1660 RETURN

-1670 IF I=1 THEN 1690
 1680 DATA SAVE DC #2,C$()
-1690 DATA SAVE DC #2,END
 1700 DBACKSPACE #1,BEG
 1710 DBACKSPACE #2,BEG
 1720 DBACKSPACE #3,BEG
 1730 INIT(20) P$()
 1740 I=1
-1750 DATA LOAD DC #2,A$()
 1760 IF END THEN 1840
 1770 IF I[]1 THEN 1800
 1780 P$(1)="A  "
 1790 GOTO 1810

-1800 P$(I)=STR(A$(1),1,3)
-1810 DATA SAVE DC #1,A$()
 1820 I=I+1
 1830 GOTO 1750

-1840 DATA SAVE DC #1,END
 1850 P$(I)="ZZZ"
 1860 DATA SAVE DC #3,P$()
 1870 DATA SAVE DC #3,END
 1880 DBACKSPACE #1,BEG
 1890 DBACKSPACE #2,BEG
 1900 DBACKSPACE #3,BEG
 1910 INIT(FF) B$()
 1920 RETURN
```

APPENDIX D

SUPPLEMENTARY WORD FILE SUBSYSTEM

Program START3 is the menu for the supplementary word file subsystem. START3 allows the operator of the readability system an opportunity of selecting 1 of 3 programs for execution. Upon selection of option 1 the program XSUPINPT is loaded and executed. XSUPINPT allows the user the means of creating a new supplementary word file. If option 2 is selected by the operator, then program XSUPEDIT is loaded and executed. XSUPEDIT permits the operator a way to edit a supplementary word file that already exists. When option 3 is selected, the program XSUPPRNT is loaded into memory and executed. XSUPPRNT prints a supplementary word file on the line printer. Selection of option 4 returns the operator to the main system menu.

```
 0010 %### SUPPLEMENTARY WORD FILE SUBSYSTEM MENU PROGRAM  START3
-0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633
 0390 IF O=99 THEN 638
 0400 O=0
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %           * * *  SUPPLEMENTARY WORD FILE SUBSYSTEM MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1     -  LOAD SUPPLEMENTARY WORD FILE INPUT PROGRAM
-0565 %     2     -  LOAD SUPPLEMENTARY WORD FILE EDIT PROGRAM
-0575 %     3     -  LOAD SUPPLEMENTARY WORD FILE PRINT PROGRAM
-0630 %    99     -  **  Return to Master Menu  **
-0631 LOAD DC T#0, "XSUPINPT"

-0632 LOAD DC T#0, "XSUPEDIT"

-0633 LOAD DC T#0, "XSUPPRNT"

-0638 LOAD DC T#0, "START" BEG 310
```

SUPPLEMENTARY WORD FILE INPUT PROGRAM (XSUPINPT)

The supplementary word file input program allows the user to create a new supplementary word file. The user can also delete words he has already entered on the new file. The supplementary word file is made up of blocks of 100 words of 20 characters each. This program requires a work area of 4000 sectors called "TEMPWORK". This work area is used to merge new blocks of words with the words already created. This program uses a block pointer table but does not save it on disk. After each block of 100 words has been entered, the program sorts the current block of words and then merges it with the words that have already been entered.

The variables are as follows:

A$(100)20 = Input/Output block for supplementary word file

A1$3 = Device address of the supplementary word file

A2$3 = Device address of the temporary work file

A3$8 = Filename of the supplementary word file

B$(100)20 = Current block of words

B1$(2)2 = Locator array for MAT SEARCH statements

C$(100)20 = Output block of words for merging the current block of words with the words already entered.

F3$(100)2 = Work area for MAT SORT of current block of words

F4$(100)2 = Locator array output from MAT SORT

P$(50)3 = Block pointer table

W1$20 = One input word

Z1$1 = Yes/No switch for word already entered

Z2$1 = Option number to delete word or leave it unchanged

Z3$1 = Yes/No switch for end of file during merge of current block of words with words already entered

```
 0005 %## PROGRAM TO INPUT SUPPLEMENTARY WORD FILES  (XSUPINPT)
 0010 DIM A$(100)20,A1$3,A2$3,A3$8,B$(100)20,P$(50)3
 0020 DIM W1$20,B1$(2)2,Z1$1,Z2$1,F3$(100)2,F4$(100)2,C$(100)20,Z3$1
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 LINPUT "ENTER THE DEVICE OF THE SUPPLEMENTARY WORD FILE",-A1$
 0130 LINPUT "ENTER THE DEVICE OF THE WORK FILE",-A2$
 0140 PRINT HEX(03)
-0150 LINPUT "ENTER THE FILENAME OF THE SUPPLEMENTARY WORD FILE",-A3$
 0160 PRINT HEX(03)
 0170 SELECT #1 [A1$], #2 [A2$]
 0180 LIMITS T#1,A3$,U1,U2,U3,S1
 0190 IF S1=2 THEN 194
 0191 IF S1=0 THEN 211
 0192 IF S1=-2 THEN 213
 0193 PRINT "INVALID FILENAME.....RE-ENTER";HEX(07)
     : GOTO 150

-0194 PRINT A3$;" IS AN ACTIVE FILE"
 0195 PRINT "EITHER SCRATCH THE FILE OR CHOSE ANOTHER NAME"
 0196 PRINT "THEN RE-RUN THIS PROGRAM"
 0197 STOP
-0211 DATA SAVE DC OPEN T$#1,(90),A3$
 0212 GOTO 220

-0213 DATA SAVE DC OPEN T$#1,(A3$),A3$
-0220 DATA LOAD DC OPEN T#1,A3$
 0222 DATA SAVE DC #1,A$()
 0224 DATA SAVE DC #1,END
 0226 LIMITS T#1,A3$,U1,U2,U3,S1
 0230 LIMITS T#2,"TEMPWORK",L1,L2,L3,S
 0240 IF S=2 THEN 330
 0250 IF S=0 THEN 300
 0260 IF S=-2 THEN 320
 0270 PRINT "TEMPWORK IS CATALOGED AS A PROGRAM FILE"
 0280 PRINT "THIS PROGRAM REQUIRES A WORK AREA CALLED TEMPWORK"
 0290 STOP "A PROGRAMMER MUST CHANGE THIS PROGRAM OR THE DISK CATALOG"
-0300 DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"
 0310 GOTO 330

-0320 DATA SAVE DC OPEN T$#2,("TEMPWORK"),"TEMPWORK"
-0330 DATA LOAD DC OPEN T#2,"TEMPWORK"
 0340 INIT(FF) B$()
 0350 P$(1)="A  "
 0360 P$(2)="ZZZ"
-0430 PRINT HEX(03)
 0440 Z2$=" "
 0450 W1$=" "
-0460 LINPUT "ENTER WORD OR EOL",-W1$
 0470 IF W1$[]" " THEN 500
 0480 PRINT HEX(07)
 0490 GOTO 460

-0500 IF W1$="EOL" THEN 850
 0510 GOSUB '110
 0520 IF Z1$="N" THEN 630
 0530 PRINT W1$;" IS ALREADY ON FILE"
-0540 LINPUT "ENTER 1) TO DELETE WORD  2) TO LEAVE WORD UNCHANGED",-Z2$
 0550 IF Z2$="1" THEN 600
 0560 IF Z2$="2" THEN 430
 0570 Z2$=" "
 0580 PRINT HEX(07)
 0590 GOTO 540

-0600 INIT(FF) A$(Q)
 0610 DATA SAVE DC #1,A$()
 0620 GOTO 430

-0630 MAT SEARCH B$(), =STR(W1$,1,20) TO B1$() STEP 20
 0640 IF B1$(1)[]HEX(0000) THEN 690
 0650 V=V+1
 0660 IF V=101 THEN GOSUB '120
 0670 B$(V)=W1$
 0680 GOTO 430

-0690 PRINT W1$;" IS ALREADY ON FILE"
-0700 LINPUT "ENTER 1) TO DELETE WORD  2) TO LEAVE WORD UNCHANGED",-Z2$
 0710 IF Z2$="1" THEN 760
 0720 IF Z2$="2" THEN 430
 0730 Z2$=" "
 0740 PRINT HEX(07)
 0750 GOTO 700

-0760 V1=INT(VAL(B1$(1),2)/20)+1
 0770 INIT(FF) B$(V1)
 0780 GOTO 430

-0850 IF V=0 THEN 880
 0860 IF V=1 AND STR(B$(1),1,2)=HEX(FFFF) THEN 880
 0870 GOSUB '120
-0880 PRINT HEX(03)
 0890 SELECT P3
 0900 FOR I=1 TO 5
 0910 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0920 NEXT I
 0930 SELECT PO
 0940 LOAD DC T#0,"START3"

 1000 DEFFN '110
 1010 Z1$="N"
 1020 MAT SEARCH P$(), ]STR(W1$,1,3) TO B1$() STEP 3
 1030 P6=INT(VAL(B1$(1),2)/3)
-1040 DATA LOAD DA T#1, (U1+(P6-1)*9) A$()
 1050 MAT SEARCH A$(), =STR(W1$,1,20) TO B1$() STEP 20
 1060 IF B1$(1)=HEX(0000) THEN 1120
 1070 Z1$="Y"
 1080 Q=INT(VAL(B1$(1),2)/20)+1
 1105 DBACKSPACE #1,BEG
 1106 DSKIP #1,((P6-1)*9)S
 1110 GOTO 1150

-1120 IF STR(W1$,1,3)[] STR(A$(1),1,3) THEN 1150
 1125 IF P6=1 THEN 1150
 1130 P6=P6-1
 1140 GOTO 1040

-1150 RETURN

 1200 DEFFN '120
 1205 Z3$="N"
 1210 MAT SORT B$() TO F3$(),F4$()
 1220 MAT MOVE B$(),F4$() TO A$()
 1230 MAT COPY A$() TO B$()
 1240 DBACKSPACE #1,BEG
 1250 PRINT HEX(03)
 1260 PRINT AT(10,10);"ONE MOMENT PLEASE WHILE I UPDATE ";A3$
 1270 I=1
 1280 J=1
 1290 K=1
 1300 DATA LOAD DC #1,A$()
-1310 IF A$(K)=" " THEN GOSUB '150
 1315 IF Z3$="Y" AND STR(B$(J),1,2)=HEX(FFFF) THEN 1580
 1320 IF STR(A$(K),1,20)[STR(B$(J),1,20) THEN 1380
 1322 IF STR(A$(K),1,2)[]HEX(FFFF) THEN 1330
 1323 IF Z3$="Y" THEN 1330
 1324 K=K+1
 1325 IF K=101 THEN GOSUB '150
 1326 GOTO 1310

-1330 C$(I)=B$(J)
 1340 I=I+1
 1350 IF I=101 THEN GOSUB '160
 1360 J=J+1
 1365 IF J[]101 THEN 1310
 1370 J=1
 1372 INIT(FF) B$()
 1375 GOTO 1310

-1380 C$(I)=A$(K)
 1390 I=I+1
 1400 IF I=101 THEN GOSUB '160
 1410 K=K+1
 1420 IF K=101 THEN GOSUB '150
 1430 GOTO 1310

 1440 DEFFN '150
 1450 DATA LOAD DC #1,A$()
 1460 IF END THEN 1490
 1470 K=1
 1480 GOTO 1520

-1490 Z3$="Y"
 1500 INIT(FF) A$()
 1510 K=1
-1520 RETURN

 1530 DEFFN '160
 1540 DATA SAVE DC #2,C$()
 1550 INIT(20)C$()
 1560 I=1
 1570 RETURN

-1580 IF I=1 THEN 1600
 1590 DATA SAVE DC #2,C$()
-1600 DATA SAVE DC #2,END
 1610 DBACKSPACE #1,BEG
 1620 DBACKSPACE #2,BEG
 1640 INIT(20) P$()
 1650 I=1
-1660 DATA LOAD DC #2,A$()
 1670 IF END THEN 1750
 1680 IF I[]1 THEN 1710
 1690 P$(1)="A  "
 1700 GOTO 1720

-1710 P$(I)=STR(A$(1),1,3)
-1720 DATA SAVE DC #1,A$()
 1730 I=I+1
 1740 GOTO 1660

-1750 DATA SAVE DC #1,END
 1755 P$(I)="ZZZ"
 1780 DBACKSPACE #1,BEG
 1790 DBACKSPACE #2,BEG
 1810 V=1
 1820 INIT(FF) B$()
 1830 RETURN
```

SUPPLEMENTARY WORD FILE EDIT PROGRAM (XSUPEDIT)

The supplementary word file edit program allows the user to edit a supplementary word file that is already on disk. The user can add or delete words. This program works exactly like the supplementary word file input program except that the supplementary word file must already be on the disk. Also, all the variables are the same as in the supplementary word file input program.

```
 0005 %### PROGRAM TO EDIT SUPPLEMENTARY WORD FILES  (XSUPEDIT)
 0010 DIM A$(100)20,A1$3,A2$3,A3$8,B$(100)20,P$(50)3
 0020 DIM W1$20,B1$(2)2,Z1$1,Z2$1,F3$(100)2,F4$(100)2,C$(100)20,Z3$1
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 LINPUT "ENTER THE DEVICE OF THE SUPPLEMENTARY WORD FILE",-A1$
 0130 LINPUT "ENTER THE DEVICE OF THE WORK FILE",-A2$
 0140 PRINT HEX(03)
-0150 LINPUT "ENTER THE FILENAME OF THE SUPPLEMENTARY WORD FILE",-A3$
 0160 PRINT HEX(03)
 0170 SELECT #1 [A1$], #2 [A2$]
 0180 LIMITS T#1,A3$,U1,U2,U3,S1
 0190 IF S1=2 THEN 220
 0200 PRINT "INVALID FILENAME.....RE-ENTER";HEX(07)
 0210 GOTO 150

-0220 DATA LOAD DC OPEN T#1,A3$
 0230 LIMITS T#2,"TEMPWORK",L1,L2,L3,S
 0240 IF S=2 THEN 330
 0250 IF S=0 THEN 300
 0260 IF S=-2 THEN 320
 0270 PRINT "TEMPWORK IS CATALOGED AS A PROGRAM FILE"
 0280 PRINT "THIS PROGRAM REQUIRES A WORK AREA CALLED TEMPWORK"
 0290 STOP "A PROGRAMMER MUST CHANGE THIS PROGRAM OR THE DISK CATALOG"
-0300 DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"
 0310 GOTO 330

-0320 DATA SAVE DC OPEN T$#2,("TEMPWORK"),"TEMPWORK"
-0330 DATA LOAD DC OPEN T#2,"TEMPWORK"
 0340 INIT(FF) B$()
 0350 P$(1)="A  "
 0360 Z9=2
 0365 DSKIP #1,1
-0370 DATA LOAD DC #1,A$()
 0380 IF END THEN 420
 0390 P$(Z9)=A$(1)
 0400 Z9=Z9+1
 0410 GOTO 370

-0420 P$(Z9)="ZZZ"
-0430 PRINT HEX(03)
 0440 Z2$=" "
 0450 W1$=" "
-0460 LINPUT "ENTER WORD OR EOL",-W1$
 0470 IF W1$[]" " THEN 500
 0480 PRINT HEX(07)
 0490 GOTO 460
```

```
-0500 IF W1$="EOL" THEN 850
 0510 GOSUB '110
 0520 IF Z1$="N" THEN 630
 0530 PRINT W1$;" IS ALREADY ON FILE"
-0540 LINPUT "ENTER 1) TO DELETE WORD  2) TO LEAVE WORD UNCHANGED",-Z2$
 0550 IF Z2$="1" THEN 600
 0560 IF Z2$="2" THEN 430
 0570 Z2$=" "
 0580 PRINT HEX(07)
 0590 GOTO 540

-0600 INIT(FF) A$(Q)
 0610 DATA SAVE DC #1,A$()
 0620 GOTO 430

-0630 MAT SEARCH B$(), =STR(W1$,1,20) TO B1$() STEP 20
 0640 IF B1$(1)[]HEX(0000) THEN 690
 0650 V=V+1
 0660 IF V=101 THEN GOSUB '120
 0670 B$(V)=W1$
 0680 GOTO 430

-0690 PRINT W1$;" IS ALREADY ON FILE"
-0700 LINPUT "ENTER 1) TO DELETE WORD  2) TO LEAVE WORD UNCHANGED",-Z2$
 0710 IF Z2$="1" THEN 760
 0720 IF Z2$="2" THEN 430
 0730 Z2$=" "
 0740 PRINT HEX(07)
 0750 GOTO 700

-0760 V1=INT(VAL(B1$(1),2)/20)+1
 0770 INIT(FF) B$(V1)
 0780 GOTO 430

-0850 IF V=0 THEN 880
 0860 IF V=1 AND STR(B$(1),1,2)=HEX(FFFF) THEN 880
 0870 GOSUB '120
-0880 PRINT HEX(03)
 0890 SELECT P3
 0900 FOR I=1 TO 5
 0910 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0920 NEXT I
 0930 SELECT P0
 0940 LOAD DC T#0,"START3"

 1000 DEFFN '110
 1010 Z1$="N"
 1020 MAT SEARCH P$(), ]STR(W1$,1,3) TO B1$() STEP 3
 1030 P6=INT(VAL(B1$(1),2)/3)
-1040 DATA LOAD DA T#1, (U1+(P6-1)*9) A$()
 1050 MAT SEARCH A$(), =STR(W1$,1,20) TO B1$() STEP 20
 1060 IF B1$(1)=HEX(0000) THEN 1120
 1070 Z1$="Y"
 1080 Q=INT(VAL(B1$(1),2)/20)+1
 1105 DBACKSPACE #1,BEG
 1106 DSKIP #1,((P6-1)*9)S
 1110 GOTO 1150

-1120 IF STR(W1$,1,3)[] STR(A$(1),1,3) THEN 1150
 1125 IF P6=1 THEN 1150
 1130 P6=P6-1
 1140 GOTO 1040

-1150 RETURN

 1200 DEFFN '120
 1205 Z3$="N"
 1210 MAT SORT B$() TO F3$(),F4$()
 1220 MAT MOVE B$(),F4$() TO A$()
 1230 MAT COPY A$() TO B$()
 1240 DBACKSPACE #1,BEG
 1250 PRINT HEX(03)
 1260 PRINT AT(10,10);"ONE MOMENT PLEASE WHILE I UPDATE ";A3$
 1270 I=1
 1280 J=1
 1290 K=1
 1300 DATA LOAD DC #1,A$()
-1310 IF A$(K)=" " THEN GOSUB '150
 1315 IF Z3$="Y" AND STR(B$(J),1,2)=HEX(FFFF) THEN 1580
 1320 IF STR(A$(K),1,20)[STR(B$(J),1,20) THEN 1380
 1322 IF STR(A$(K),1,2)[]HEX(FFFF) THEN 1330
 1323 IF Z3$="Y" THEN 1330
 1324 K=K+1
 1325 IF K=101 THEN GOSUB '150
 1326 GOTO 1310

-1330 C$(I)=B$(J)
 1340 I=I+1
 1350 IF I=101 THEN GOSUB '160
 1360 J=J+1
 1365 IF J[]101 THEN 1310
 1370 J=1
 1372 INIT(FF) B$()
 1375 GOTO 1310

-1380 C$(I)=A$(K)
 1390 I=I+1
 1400 IF I=101 THEN GOSUB '160
 1410 K=K+1
 1420 IF K=101 THEN GOSUB '150
 1430 GOTO 1310

 1440 DEFFN '150
 1450 DATA LOAD DC #1,A$()
 1460 IF END THEN 1490
 1470 K=1
 1480 GOTO 1520

-1490 Z3$="Y"
 1500 INIT(FF) A$()
 1510 K=1
-1520 RETURN

 1530 DEFFN '160
 1540 DATA SAVE DC #2,C$()
 1550 INIT(20)C$()
 1560 I=1
 1570 RETURN

-1580 IF I=1 THEN 1600
 1590 DATA SAVE DC #2,C$()
-1600 DATA SAVE DC #2,END
 1610 DBACKSPACE #1,BEG
 1620 DBACKSPACE #2,BEG
 1640 INIT(20) P$()
 1650 I=1
-1660 DATA LOAD DC #2,A$()
 1670 IF END THEN 1750
 1680 IF I[]1 THEN 1710
 1690 P$(1)="A  "
 1700 GOTO 1720

-1710 P$(I)=STR(A$(1),1,3)
-1720 DATA SAVE DC #1,A$()
 1730 I=I+1
 1740 GOTO 1660

-1750 DATA SAVE DC #1,END
 1755 P$(I)="ZZZ"
 1780 DBACKSPACE #1,BEG
```

```
1790 DBACKSPACE #2,BEG
1810 V=1
1820 INIT(FF) B$()
1830 RETURN
```

SUPPLEMENTARY WORD FILE PRINT PROGRAM (XSUPPRNT)

The supplementary word file print program allows the user to print a supplementary word file. It is a simple disk to print program with the option to print a title. The variables in this program are as follows:

A\$(100)20 = Input block of words

A1\$3 = Device address where the supplementary word file is stored

A2\$8 = Filename of supplementary word file

A3\$3 = Device address of the printer

A4\$40 = Title for printout

Z1\$1 = Yes/No switch for title

```
 0010 %## PROGRAM ]  SUPPLEMENTARY WORD FILE LIST   (XSUPPRNT)
 0020 DIM A$(100)20,A1$3,A2$8,A3$3,Z1$1,A4$40
 0110 SELECT PRINT 005(80)
 0120 PRINT HEX(03)
 0130 LINPUT "ENTER DEVICE FOR FILE STORAGE",-A1$
 0135 SELECT #1 [A1$]
-0140 LINPUT "ENTER FILENAME OF SUPPLEMENTARY WORD FILE",-A2$
 0144 LIMITS T#1,STR(A2$,1,8),L1,L2,L3,S
 0145 IF S=2 THEN 150
 0146 PRINT "INVALID FILENAME...."
 0147 A2$=" "
 0148 GOTO 140

-0150 LINPUT "ENTER DEVICE FOR PRINTING LIST",-A3$
 0160 PRINT HEX(03)
 0162 LINPUT "DO YOU WISH TO TITLE THIS LIST? (Y/N)",-Z1$
 0163 IF Z1$="N" THEN 169
 0164 LINPUT "ENTER TITLE ",-A4$
-0169 PRINT HEX(03)
 0170 PRINT AT(10,10);"PRINTING FILE NOW...."
 0175 SELECT PRINT [A3$] (132)
 0190 DATA LOAD DC OPEN T#1,A2$
-0200 DATA LOAD DC #1,A$()
 0205 IF END THEN 420
 0210 PRINT HEX(0C)
 0230 C8=INT((90-LEN(A4$)*2)/4)
 0240 PRINT TAB(C8);HEX(0E);A4$
 0250 PRINT
 0260 FOR I=1 TO 50
 0280 IF A$(I)=" " THEN 310
 0290 PRINTUSING 390,A$(I),A$(I+50),A$(I+100),A$(I+150)
-0310 NEXT I
 0320 GOTO 200

-0390 %####################     ####################     ####################
          ####################
-0420 PRINT HEX(0C)
 0425 SELECT PRINT 005(80)
 0430 PRINT HEX(03)
 0440 SELECT P3
 0450 FOR I=1 TO 5
 0460 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0470 NEXT I
 0480 SELECT P0
 0490 LOAD DC T#0,"START3"
```

APPENDIX E

PHRASE SUBSTITUTION FILE SUBSYSTEM

Program START4 is the menu for the phrase substitution file subsystem. There are FOUR options for the user at the time START4 is loaded into memory and execution has been initiated. Option 1 loads and initializes the program XPHRINPT. This program is the one used to create a new phrase substitution file. Option 2 loads and initializes the execution of the program XPHREDIT. This program permits the user a means of editing a phrase substitution file that already exists on disk. If option 3 is selected, then the program XPHRPRNT is executed. XPHRPRNT prints a hard copy of a selected phrase substitution file. Option 4 is selected to terminate the execution of START4 and to return control to the master menu program.

```
 0010 %## PHRASE SUBSTITUTION FILE SUBSYSTEM MENU PROGRAM  START4
-0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633
 0390 IF O=99 THEN 638
 0400 O=0
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %           * * *  PHRASE SUBSTITUTION SUBSYSTEM MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1     -  LOAD PHRASE SUBSTITUTION FILE INPUT PROGRAM
-0565 %     2     -  LOAD PHRASE SUBSTITUTION FILE EDIT PROGRAM
-0575 %     3     -  LOAD PHRASE SUBSTITUTION FILE PRINT PROGRAM
-0630 %    99     -  ** Return to Master Menu **
-0631 LOAD DC T#0, "XPHRINPT"

-0632 LOAD DC T#0, "XPHREDIT"

-0633 LOAD DC T#0, "XPHRPRNT"

-0638 LOAD DC T#0, "START" BEG 310
```

PHRASE SUBSTITUTION FILE INPUT PROGRAM (XPHRINPT)

The phrase substitution file input program allows the user to create a new phrase substitution file. The user can also delete phrases or change substitutes of phrases he has already entered on the new file. A phrase substitution file consists of one block of 126 records of 80 characters each. The records are broken down into three parts. Characters 1-30 are the original phrase, characters 31-55 are the first suggested substitute, and characters 56-80 are the second suggested substitute. The second substitute may be blank. The program sorts the table of phrases alphabetically on characters 1-30 after each new phrase is added.

The variables used in this program are as follows:

A\$(126)80 = Table of phrases and their substitutes

A1\$3 = Device address where the phrase substitution file is stored

A2\$8 = Filename of the phrase substitution file

B1\$(2)2 = Locator array for MAT SEARCH

C\$(126)80 = Output area for MAT MOVE of A\$()

F3\$(126)2 = Work area for MAT SORT of A\$()

F4\$(126)2 = Output locator array for MAT SORT of A\$()

P1\$30 = One input phrase

P2\$25 = First substitute for original phrase

P3\$25 = Second substitute

Z1\$1 = Yes/No switch for phrase table full

Z2\$1 = Option number to delete, change substitutes, or leave phrase alone

```
 0005 %## PROGRAM TO INPUT PHRASE SUBSTITUTION FILE  (XPHRINPT)
 0010 DIM A$(126)80,A1$3,A2$8,P1$30,P2$25,P3$25
 0020 DIM Z1$1,Z2$1,B1$(2)2,F3$(126)2,F4$(126)2,C$(126)80
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 LINPUT "ENTER THE DEVICE OF THE PHRASE SUBSTITUTION FILE",-A1$
-0130 LINPUT "ENTER THE FILENAME OF THE PHRASE SUBSTITUTION FILE",-A2$
 0140 SELECT #1 [A1$]
 0150 LIMITS T#1,A2$,L1,L2,L3,S
 0160 IF S=2 THEN 210
 0170 IF S=0 THEN 250
 0180 IF S=-2 THEN 270
 0190 PRINT "INVALID FILENAME.....RE-ENTER";HEX(07)
 0200 GOTO 130

-0210 PRINT A2$;" IS AN ACTIVE FILE"
 0220 PRINT "EITHER SCRATCH THE FILE OR CHOSE ANOTHER NAME"
 0230 PRINT "THEN RE-START THIS PROGRAM"
 0240 STOP
-0250 DATA SAVE DC OPEN T$#1,(45),A2$
 0260 GOTO 280

-0270 DATA SAVE DC OPEN T$#1,(A2$),A2$
-0280 DATA LOAD DC OPEN T#1,A2$
-0290 PRINT HEX(03)
 0300 INIT(20) P1$,P2$,P3$
 0310 Z2$=" "
-0320 LINPUT "ENTER PHRASE OR EOL",-P1$
 0330 IF P1$[]" " THEN 360
 0340 PRINT HEX(07)
 0350 GOTO 320

-0360 IF P1$="EOL" THEN 660
 0370 MAT SEARCH A$(), =STR(P1$,1,30) TO B1$() STEP 80
 0380 IF B1$(1)=HEX(0000) THEN 590
 0390 Q=INT(VAL(B1$(1),2)/80)+1
 0400 PRINT
 0410 PRINT "PHRASE IS ALREADY ON FILE"
-0420 PRINT "ENTER:"
 0421 PRINT "(1) TO DELETE PHRASE"
 0422 PRINT "(2) TO CHANGE SUBSTITUTES"
 0423 PRINT "(3) TO LEAVE PHRASE UNCHANGED"
 0424 LINPUT -Z2$
 0430 IF Z2$="3" THEN 290
 0440 IF Z2$="2" THEN 520
 0450 IF Z2$="1" THEN 490
 0460 Z2$=" "
 0470 PRINT HEX(07)
 0480 GOTO 420

-0490 INIT(FF) A$(Q)
 0500 GOSUB '110
 0510 GOTO 290

-0520 P2$=STR(A$(Q),31,25)
 0530 P3$=STR(A$(Q),56,25)
 0540 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-P2$
 0550 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-P3$
 0560 STR(A$(Q),31,25)=P2$
 0570 STR(A$(Q),56,25)=P3$
 0580 GOTO 290

-0590 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-P2$
 0600 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-P3$
 0610 STR(A$(126),1,30)=P1$
 0620 STR(A$(126),31,25)=P2$
 0630 STR(A$(126),56,25)=P3$
 0640 GOSUB '110
 0650 GOTO 290

-0660 DATA SAVE DC #1,A$()
 0720 DATA SAVE DC #1,END
 0730 IF Z1$="Y" THEN STOP
 0740 PRINT HEX(03)
 0750 SELECT P3
 0760 FOR I=1 TO 5
 0770 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0780 NEXT I
 0790 SELECT P0
 0800 LOAD DC T#0,"START4"

 0810 DEFFN '110
 0820 Z1$="N"
 0830 FOR I=1 TO 126
 0840 IF A$(I)[]" " THEN 860
 0850 INIT(FF) A$(I)
-0860 NEXT I
 0870 MAT SORT A$() (1,30) TO F3$(),F4$()
 0880 MAT MOVE A$(),F4$() TO C$()
 0890 MAT COPY C$() TO A$()
 0900 FOR I=1 TO 126
 0910 IF STR(A$(I),1,2)[]HEX(FFFF) THEN 930
 0920 INIT(20) A$(I)
-0930 NEXT I
```

```
0940 IF A$(126)=" " THEN RETURN
0950 PRINT HEX(03)
0960 PRINT "THE PHRASE TABLE IS NOW FULL"
0970 PRINT "THE LAST PHRASE TO BE ENTERED WAS ";P1$
0980 PRINT "IF YOU WISH TO ADD MORE PHRASES WITHOUT DELETING SOME FIRST"
0990 PRINT "THEN CONTACT A PROGRAMMER"
1000 Z1$="Y"
1010 GOTO 660
```

PHRASE SUBSTITUTION FILE EDIT PROGRAM (XPHREDIT)

The phrase substitution file edit program allows the user to edit a phrase substitution file that already exists on disk. The user can add or delete phrases or he can change substitutes for phrases already on file. This program works exactly the same way as the phrase substitution input program except that the phrase substitution file must already be on the disk. The variables and their uses are the same as for the phrase substitution input program.

```
 0005 %## PROGRAM TO EDIT PHRASE SUBSTITUTION FILE  (XPHREDIT)
 0010 DIM A$(126)80,A1$3,A2$8,P1$30,P2$25,P3$25
 0020 DIM Z1$1,Z2$1,B1$(2)2,F3$(126)2,F4$(126)2,C$(126)80
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 LINPUT "ENTER THE DEVICE OF THE PHRASE SUBSTITUTION FILE",-A1$
-0130 LINPUT "ENTER THE FILENAME OF THE PHRASE SUBSTITUTION FILE",-A2$
 0140 SELECT #1 [A1$]
 0150 LIMITS T#1,A2$,L1,L2,L3,S
 0160 IF S=2 THEN 280
 0190 PRINT "INVALID FILENAME.....RE-ENTER";HEX(07)
 0200 GOTO 130

-0280 DATA LOAD DC OPEN T#1,A2$
 0285 DATA LOAD DC #1,A$()
 0286 DBACKSPACE #1,BEG
-0290 PRINT HEX(03)
 0300 INIT(20) P1$,P2$,P3$
 0310 Z2$=" "
-0320 LINPUT "ENTER PHRASE OR EOL",-P1$
 0330 IF P1$[]" " THEN 360
 0340 PRINT HEX(07)
 0350 GOTO 320

-0360 IF P1$="EOL" THEN 660
 0370 MAT SEARCH A$(), =STR(P1$,1,30) TO B1$() STEP 80
 0380 IF B1$(1)=HEX(0000) THEN 590
 0390 Q=INT(VAL(B1$(1),2)/80)+1
 0400 PRINT
 0410 PRINT "PHRASE IS ALREADY ON FILE"
-0420 PRINT "ENTER:"
 0421 PRINT "(1) TO DELETE PHRASE"
 0422 PRINT "(2) TO CHANGE SUBSTITUTES"
 0423 PRINT "(3) TO LEAVE PHRASE UNCHANGED"
 0424 LINPUT -Z2$
 0430 IF Z2$="3" THEN 290
 0440 IF Z2$="2" THEN 520
 0450 IF Z2$="1" THEN 490
 0460 Z2$=" "
 0470 PRINT HEX(07)
 0480 GOTO 420

-0490 INIT(FF) A$(Q)
 0500 GOSUB '110
 0510 GOTO 290

-0520 P2$=STR(A$(Q),31,25)
```

```
 0530 P3$=STR(A$(Q),56,25)
 0540 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-P2$
 0550 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-P3$
 0560 STR(A$(Q),31,25)=P2$
 0570 STR(A$(Q),56,25)=P3$
 0580 GOTO 290

-0590 LINPUT "ENTER FIRST SUBSTITUTE OR C/R",-P2$
 0600 LINPUT "ENTER SECOND SUBSTITUTE OR C/R",-P3$
 0610 STR(A$(126),1,30)=P1$
 0620 STR(A$(126),31,25)=P2$
 0630 STR(A$(126),56,25)=P3$
 0640 GOSUB '110
 0650 GOTO 290

-0660 DATA SAVE DC #1,A$()
 0720 DATA SAVE DC #1,END
 0730 IF Z1$="Y" THEN STOP
 0740 PRINT HEX(03)
 0750 SELECT P3
 0760 FOR I=1 TO 5
 0770 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0780 NEXT I
 0790 SELECT P0
 0800 LOAD DC T#0,"START4"

 0810 DEFFN '110
 0820 Z1$="N"
 0830 FOR I=1 TO 126
 0840 IF A$(I)[]" " THEN 860
 0850 INIT(FF) A$(I)
-0860 NEXT I
 0870 MAT SORT A$() (1,30) TO F3$(),F4$()
 0880 MAT MOVE A$(),F4$() TO C$()
 0890 MAT COPY C$() TO A$()
 0900 FOR I=1 TO 126
 0910 IF STR(A$(I),1,2)[]HEX(FFFF) THEN 930
 0920 INIT(20) A$(I)
-0930 NEXT I
 0940 IF A$(126)=" " THEN RETURN
 0950 PRINT HEX(03)
 0960 PRINT "THE PHRASE TABLE IS NOW FULL"
 0970 PRINT "THE LAST PHRASE TO BE ENTERED WAS ";P1$
 0980 PRINT "IF YOU WISH TO ADD MORE PHRASES WITHOUT DELETING SOME FIRST"
 0990 PRINT "THEN CONTACT A PROGRAMMER"
 1000 Z1$="Y"
 1010 GOTO 660
```

PHRASE SUBSTITUTION FILE PRINT PROGRAM (XPHRPRNT)

The phrase substitution file print program allows the user to print a phrase substitution file. This is just a simple disk-to-print program with headings and an optional title.

The variables used in this program are as follows:

A$(126)80 = Input area for phrase substitution file

A1$3 = Device address where phrase substitution file is stored

A2$8 = Filename of phrase substitution file

A3$3 = Device address of printer

A4$40 = Optional title

W1$30 = Original phrase

W2$25 = First suggested substitute

W3$25 = Second suggested substitute

```
 0010 %## PROGRAM ]  PHRASE SUBSTITUTION FILE LIST   (XPHRPRNT)
 0020 DIM A$(126)80,A1$3,A2$8,A3$3,Z1$1,A4$40
 0030 DIM W1$30,W2$25,W3$25
 0100 L1=99
 0110 SELECT PRINT 005(80)
 0120 PRINT HEX(03)
 0130 LINPUT "ENTER DEVICE FOR FILE STORAGE",-A1$
 0135 SELECT #1 [A1$]
-0140 LINPUT "ENTER FILENAME OF PHRASE SUBSTITUTION FILE",-A2$
 0144 LIMITS T#1,STR(A2$,1,8),L1,L2,L3,S
 0145 IF S=2 THEN 150
 0146 PRINT "INVALID FILENAME...."
 0147 A2$=" "
 0148 GOTO 140

-0150 LINPUT "ENTER DEVICE FOR PRINTING LIST",-A3$
 0160 PRINT HEX(03)
 0162 LINPUT "DO YOU WISH TO TITLE THIS LIST? (Y/N)",-Z1$
 0163 IF Z1$="N" THEN 169
 0164 LINPUT "ENTER TITLE ",-A4$
-0169 PRINT HEX(03)
 0170 PRINT AT(10,10);"PRINTING FILE NOW...."
 0175 SELECT PRINT [A3$] (132)
 0190 DATA LOAD DC OPEN T#1,A2$
 0200 DATA LOAD DC #1,A$()
 0210 FOR I=1 TO 126
 0220 IF A$(I)=" " THEN 380
 0225 IF L1]55 THEN 290
-0230 W1$=STR(A$(I),1,30)
 0240 W2$=STR(A$(I),31,25)
 0250 W3$=STR(A$(I),56,25)
 0260 PRINTUSING 390,W1$,W2$,W3$
 0270 L1=L1+1
 0280 GOTO 380

-0290 PRINT HEX(0C)
 0300 L1=5
 0310 C8=INT((80-LEN(A4$)*2)/4)
 0320 PRINT TAB(C8);HEX(0E);A4$
 0330 PRINT
 0340 PRINTUSING 400
 0350 PRINTUSING 410
 0360 PRINTUSING 400
 0370 PRINT
 0375 GOTO 230

-0380 NEXT I
```

```
-0390  %   ##############################     #########################     ##
      #######################
-0400   %=====================================================================
      =========================
-0410  %       PHRASE                         FIRST SUBSTITUTE              SE
      COND SUBSTITUTE
 0420 SELECT PRINT 005(80)
 0430 PRINT HEX(03)
 0440 SELECT P3
 0450 FOR I=1 TO 5
 0460 PRINT AT(10,10);"NOW LOADING SUBSYSTEM MENU"
 0470 NEXT I
 0480 SELECT P0
 0490 LOAD DC T#0,"START4"
```

APPENDIX F

TEXT ANALYSIS PROGRAMS

Program START5 is the menu for the text analysis programs. Two options are provided for analyzing a selected text passage. Option 1 loads and executes program XXANAL04. This is the main text analysis program which reads a text passage, analyzes it according to the instructions supplied by the operator, and prints the analyzed results. Option 2 loads and executes program XXANAL01. This program is of an experimental nature and has yet to be released. Selection of option 99 returns the system to the Master Menu.

```
 0010 %## TEXT ANALYSIS SUBSYSTEM MENU PROGRAM  START5
-0310 SELECT PRINT 005(80)
    : PRINT HEX(03)
 0320 PRINTUSING 530
    : PRINT
    : PRINTUSING 540
    : PRINT
    : PRINT
    : PRINTUSING 560
    : PRINTUSING 565
    : PRINTUSING 575
 0330 PRINT
    : PRINTUSING 630
 0340 PRINT
-0360 INPUT " ENTER OPTION ",O
 0370 ON O GOTO 631,632,633
 0390 IF O=99 THEN 638
 0400 O=0
 0410 PRINT HEX(07)
 0420 GOTO 360

-0530 %              * * *  TEXT ANALYSIS SUBSYSTEM MENU  * * *
-0540 %   OPTIONS ]
-0560 %     1     -  LOAD WORD AND PHRASE ANALYSIS PROGRAM
-0565 %     2     -  **  FOR FUTURE USE  **
-0575 %     3     -  LOAD WORD AND PHRASE ANALYSIS TEST PROGRAM
-0630 %    99     -  **  Return to Master Menu  **
-0631 LOAD DC T#0, "XXANAL04"

-0632 GOTO 360

-0633 LOAD DC T#0,"XXANAL01"

-0638 LOAD DC T#0, "START" BEG 310
```

TEXT ANALYSIS PROGRAM (XXANAL04)

The text analysis program is designed to read a text passage or a whole text file and print out the analyzed results. The analysis program produces output which consists of the following:

1) Suggested substitutes for awkward words or phrases
2) Square brackets around uncommon words
3) The number of words in a sentence at the end of each sentence with more than 22 words
4) The reading grade level of the text
5) A list of all uncommon words and the number of times each one occurred

The program reads and analyzes the text one word at a time. A word is defined as a string of characters between two spaces. The program is broken down into three major sections. They are: Phrase Analysis, Word Analysis, and Print Routine.

PHRASE ANALYSIS:

The phrase analysis section has two major subroutines which are: the Get Next Word Subroutine and Check Phrase Substitution Subroutine.

The get next word subroutine reads the next word from the text and returns it to the phrase analysis section. Before it returns the word of text it makes a new word like the original except that the new word is all in upper case letters and it doesn't have any punctuation marks. When the punctuation is removed a check is made for the end of a sentence. If a period is found the program checks a list of common abbreviations so that the end of sentence switch is not turned on by a period after an abbreviation. The get next word subroutine also returns a count of the number of syllables in the word. The number of syllables is defined as being equal to the number of vowels in the word. Consecutive vowels are counted as just one and the endings "ed"and "es" are not counted. The ending "e" is not counted unless it is "le". The get next word subroutine also has a pointer to keep track of where the current word is in the text.

The check phrases subroutine compares a phrase from the text with the phrase substitution table and decides whether the text phrase is a partial phrase, a whole phrase, or no phrase at all. If the text phrase matches in content but is shorter than the table phrase then it is a partial phrase. For example, "as to" is a partial phrase of "as to whether". If a text phrase matches a table phrase in content and length then it is a whole phrase.

The phrase analysis section starts by getting a word of text. This is the first word of a possible phrase. If the word ends with punctuation, starts with a capital letter, or is a number then it can not be part of a phrase and the phrase analysis section ends. If the word can be part of a phrase then the phrase analysis section gets the next word of text and adds it to the previous word. It then checks this text phrase against the phrase table. The phrase analysis section keeps adding words and checking the phrase against the phrase table until there is not a match. If the whole phrase switch was set on during this process then the text phrase is put in angle brackets < > and the suggested substitutes are added in brackets and asterisks <* *>. If there was not a match with a whole phrase then the first word is returned to the word analysis section. The phrase analysis section also keeps track of the number of words processed, the number of sentences, the number of syllables, and the end of file.

WORD ANALYSIS:

The word analysis section receives a single word from the phrase analysis section and decides what kind of notation, if any, should go with the word. First, the word analysis section compares the word with the words in the most common words table. The most common words table is a table of the 148 most commonly used words that don't need a substitute. About half the words used in an ordinary text passage will be found on this table. If the word analysis section matches the word from the text with a word on the most common words table then nothing is done to the word and it is sent to the print routine. If the word is not in the most common words table then it is checked against the word substitution file. If it is found on the word substitution file and there are no suggested substitutes then nothing is done to the word and it is sent to the print routine. If it is found and there are suggested substitutes then the word is put in angle brackets< > and either one or both substitutes are added in angle brackets with asterisks <* *>. If the word is not found on the word substitution file and the supplementary word file option is selected then the word is checked against the supplementary word file. If it is found on the supplementary word file then nothing is done and the word is sent to the print routine. If the word is not found on any of the word lists then it is considered an uncommon word. It is then placed in square brackets [] and sent to the print routine. It is also added to the uncommon words table if it is not already there. If it is already in the uncommon words table then 1 is added to the frequency associated with that word in the table.

If a word begins with a capital letter and it is not the first word in a sentence then it is assumed to be a proper noun and the word analysis section is not executed for that word. The word analysis section is also skipped if the word from the text begins with a number.

If the word analyzed by the word analysis section happens to be the last word in a sentence with more than 22 words then the number of words in that sentence is put between dollar signs $$ $$ and added to the text being sent to the print routine.

PRINT ROUTINE:

The print routine receives text from the phrase and word analysis sections and prints it out. The print routine will print an optional title at the beginning of the printout and it will skip to a new page and print a chapter heading whenever a title in the original text is enclosed in angle brackets< >. The print routine will also print page numbers on the output.

When the end of file is reached the print routine will calculate the readability results and print them. It will also sort the uncommon words table alphabetically and print it out.

DEFINITIONS OF VARIABLES USED IN TEXT ANALYSIS PROGRAM (XXANAL04)

A$3 = Device for printing analysis

A0$8 = Filename of text file

A2$3 = Device for word substitution file

A3$3 = Device for phrase substitution file

A4$3 = Device for the text file

A6$8 = Filename of the word substitution file

A7$8 = Filename of the phrase substitution file

A8$3 = Device for supplementary word file

A9$8 = Filename of the supplementary word file

B$(3)80 = Text file input buffer

B0 = Pointer to current position in text

B1 = Pointer to temporary position in text

B1$(2)2 = Locator variable for MAT SEARCH statements

B6$8 = Filename of block pointer file

C1$40 = Title for print output

C2$40 = Table of consonants

D1$(3)80 = Three lines of text currently being processed

F = Function used to define frequency of uncommon word

F1 = Reading grade level of text

F2$(500)8 = Table of text filenames

F3 = Average number of words per sentence

F4 = Average number of syllables per word

H1$1 = Yes/No switch for heading

I = Index in FOR/NEXT loops

J = Index to next line in text - D1$(J)

K = Index to table of text filenames - F2$(K)

L1 = Starting sector address of text file

L1$54 = Table of upper and lower case letters

L2 = Ending sector address of text file

L2$90 = Line of text to print

L3 = Current sector address of text file

L4 = Line counter for print output

M0 = Pointer to end of current three lines of text = 241

P1$(126)80 = Phrase substitution table

P3 = Index to matching phrase in phrase substitution table

P4 = Index to matching word in word substitution file

P5 = Pointer to end of uncommon words table

P5$(4)1 = Table of end of sentence punctuation

P6 = Block number for word substitution file or supplementary word file

P6$(9)1 = Table of punctuation

P7 = Page count for print output

S = Status of text file

S1 = Number of sentences

S2 = Number of syllables

S3 = Words per sentence counter

T1$30 = Word of text

T2 = Syllable count for one word

T2$20 = Word of text in upper case minus punctuation

T3 = Words per phrase for partial phrase

T3$80 = Word or phrase of text plus flags and substitutes

T4 = Words per phrase for whole phrase

T4$30 = Partial phrase in upper case minus punctuation

T5 = Syllable count for partial phrase

T5$20 = First word of possible phrase in upper case minus punctuation

T6 = Syllable count for whole phrase

T6$30 = Partial phrase of text

T7$30 = Whole phrase of text

T8$(10)4 = Array for unpacked syllables

T9$20 = Upper case word used to calculate vowel count

U1$(300)21 = Table of uncommon words

U4 = Starting sector for word substitution file

U5 = Ending sector for word substitution file

U6 = Current sector for word substitution file

U7 = Starting sector for supplementary word file

U8 = Ending sector for supplementary word file

U9 = Current sector for supplementary word file

V1$2 = Words per sentence converted to alpha-numeric

V1$(300)2 = Work area for MAT SORT of uncommon words table

V2$(300)2 = Locator array for MAT SORT of uncommon words table

V3 = Index to sorted uncommon words table

V4 = Index to sorted uncommon words table

W1 = Total number of words

W3$(100)20 = Input buffer for supplementary word file

W4$(50)3 = Block pointer table for supplementary word file

W5$(200)60 = Input buffer for word substitution file

W6$(100)3 = Block pointer table for word substitution file

X1$(150)10 = Table of most common words

Y1$1 = Yes/No switch for partial phrase

Y2$1 = Yes/No switch for whole phrase

Y3$1 = Yes/No switch for first word in a phrase

Y4$1 = Yes/No switch for last line of text

Y5$1 = Yes/No switch for first word of sentence

Y6$1 = Yes/No switch for most common word

Y7$1 = Yes/No switch for text word on word substitution file

Y8$1 = Yes/No switch to suppress analysis

Y9$1 = Yes/No switch for multiple passages

Z0$1 = Input variable to continue with program after identification msg

Z1$1 = Yes/No switch for using supplementary word file

Z2$1 = Number of substitutes to print

Z3$1 = Yes/No switch for upper case text only

Z4$1 = Yes/No switch for end of file

Z5$1 = Yes/No switch for end of sentence

Z6$1 = Yes/No switch for phrase analysis

Z7$1 = Yes/No switch for word analysis

Z8$1 = Yes/No switch for title

Z9 = Counter for building supplementary word file block pointer table

DEFINITION OF SYMBOLS USED IN THE WORD AND PHRASE ANALYSIS PROGRAM

<www>	Where www is a word or phrase to be substituted
<*www/xxx*>	Where www is the first substitute and xxx is second substitute
[www]	Where www is an uncommon word
$$nn$$	Where nn is the number of words in the sentence

```
 0010 %## PROGRAM ] KINCAID READABILITY VERSION 2.0  2200MVP   5/27/80
 0020 DIM A0$8,A2$3,A3$3,A4$3,A6$8,A7$8,A8$3,A9$8,Z0$,H1$1
 0030 DIM Z1$1,A$3,C1$40,Z2$1,Z3$1,X1$(150)10
 0040 DIM W6$(100)3,W5$(200)60,P1$30,P2$(126)80
 0050 DIM W3$(100)20,W4$(50)3,P5$(4)1,P6$(8)1,L1$54,Z4$1,Z5$1,Z6$1,Z7$1
 0060 DIM Z8$1,B1$(2)2,T1$30,T2$20,T3$80,T4$30,T6$30,T7$30
 0070 DIM Y1$1,Y2$1,Y3$1,V1$2,T5$20,B$(3)80,D1$(3)80,Y4$1,Y5$1,T9$20
 0080 DIM T8$(10)4,C2$42,Y6$1,Y7$1,U1$(300)21,L2$90,V1$(300)2,V2$(300)2,Y8$1
 0090 DIM Y9$1,F2$(500)8,B6$8,X2$(40)8
 0100 SELECT PRINT 005(80)
 0110 PRINT HEX(03)
 0120 PRINTUSING 220
 0130 PRINTUSING 230
 0140 PRINTUSING 240
 0150 PRINTUSING 250
 0160 PRINTUSING 260
 0170 PRINTUSING 270
 0180 PRINTUSING 280
 0190 PRINT AT(10,1);
 0200 PRINTUSING 290," ";
 0210 INPUT Z0$
-0220 %Program  ] READABILITY PROGRAM  (ANALYSIS)
-0230 % Version  ] 2.0 - 5/27/80
-0240 % Abstract ]
-0250 %             This program will compute a readability index for
-0260 %             a passage of text.  The index calculated is based
-0270 %             on the KINCAID INDEX.  This program is a modification
-0280 %             of the GM SERVICE RESEARCH READABILITY PROGRAM.
-0290 %  Touch  'RETURN(EXEC)'  to Continue with Program ... #
 0300 PRINT HEX(03)
 0305 A2$="D33"
    : A3$="D33"
    : A4$="D33"
 0330 PRINT AT(7,2);"ENTER THE DEVICE FOR WORD SUBSTITUTION LIST";
 0340 LINPUT -A2$
 0350 PRINT AT(9,2);"ENTER THE DEVICE FOR PHRASE SUBSTITUTION LIST";
 0360 LINPUT -A3$
 0370 PRINT AT(11,2);"ENTER THE DEVICE FOR THE TEXT";
 0380 LINPUT -A4$
 0390 SELECT #7 [A2$]
 0400 SELECT #2 [A2$]
 0410 SELECT #3 [A3$]
 0420 SELECT #4 [A4$]
 0425 SELECT #6 [A4$]
 0430 PRINT HEX(03)
 0435 A6$="WORDSUB1"
    : A7$="PHRSTBL1"
```

```
   : Z2$="1"
 0460 PRINT AT(7,2);"ENTER THE FILENAME OF WORD SUBSTITUTION LIST";
 0470 LINPUT -A6$
 0480 PRINT AT(9,2);"ENTER THE FILENAME OF PHRASE SUBSTITUTION LIST";
 0490 LINPUT -A7$
 0494 B6$="BLKPOINT"
 0495 PRINT AT(11,2);"ENTER FILENAME OF BLOCK POINTER FILE";
 0496 LINPUT -B6$
 0500 Z1$="N"
 0510 PRINT HEX(03)
 0520 PRINT AT(5,2);"DO YOU WISH TO USE A SUPPLEMENTARY WORD LIST";
 0530 LINPUT -Z1$
 0540 IF Z1$="N" THEN 600
 0550 PRINT AT(7,2);"ENTER THE DEVICE OF SUPPLEMENTARY WORD LIST";
 0560 LINPUT -A8$
 0570 PRINT AT(9,2);"ENTER THE FILENAME OF SUPPLEMENTARY WORD LIST";
 0580 LINPUT -A9$
-0600 PRINT HEX(03)
 0610 PRINT AT(10,1);"ONE MOMENT PLEASE WHILE I PREPARE MYSELF"
 0615 REM INITIALIZE MOST COMMON WORDS TABLE
 0620 GOSUB 7000
 0730 REM        LOAD BLOCK POINTER ARRAY FOR WORD SUBSTITUTION FILE
 0740 DATA LOAD DC OPEN T#2,STR(A6$,1,8)
 0750 LIMITS T#2,U4,U5,U6
 0755 DATA LOAD DC OPEN T#7,B6$
 0780 DATA LOAD DC #7,W6$()
 0840 REM         LOAD PHRASE SUBSTITUTION FILE TO A TABLE
 0850 DATA LOAD DC OPEN T#3,STR(A7$,1,8)
 0870 DATA LOAD DC #3,P2$()
 0950 IF Z1$="N" THEN 1080
 0960 REM          BUILD BLOCK POINTER ARRAY FOR SUPPLEMENTARY WORD LIST
 0970 SELECT #5 [A8$]
 0980 DATA LOAD DC OPEN T#5,STR(A9$,1,8)
 0990 LIMITS T#5,U7,U8,U9
 1000 W4$(1)="A  "
 1010 Z9=2
-1020 DATA LOAD DC #5,W3$()
 1030 IF END THEN 1070
 1040 W4$(Z9)=W3$(1)
 1050 Z9=Z9+1
 1060 GOTO 1020

 1065 REM INITIALIZE VARIABLES AND SWITCHES
-1070 W4$(Z9)="ZZZ"
-1080 C2$=" B C D F G H J K L M N P Q R S T V W X Z -"
 1140 P5$(1)="."
 1150 P5$(2)="?"
 1160 P5$(3)="!"
 1165 P5$(4)=";"
 1170 P6$(1)=":"
 1190 P6$(3)=","
 1200 P6$(4)=HEX(22)
 1210 P6$(5)="'"
 1220 P6$(6)="("
 1230 P6$(7)=")"
 1240 P6$(8)="["
 1250 P6$(2)="]"
 1260 L1$="AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz"
 1270 M0=241
-1275 SELECT PRINT 005(80)
 1280 PRINT HEX(03)
 1285 C1$=" "
-1290 PRINT
 1300 A0$=" "
 1310 PRINT "ENTER FILENAME OF TEXT TO ANALYZE OR  TO STOP";
 1320 LINPUT -A0$
 1330 IF A0$="END" THEN 1730
 1340 LIMITS T#4, STR(A0$,1,8),L1,L2,L3,S
 1350 IF S=2 THEN 1390
 1360 PRINT HEX(030707)
 1370 PRINT "INVALID FILENAME......RE-ENTER"
 1380 GOTO 1290

-1390 Z8$="N"
 1391 K=0
 1400 PRINT AT(5,2);"DO YOU WANT TO TITLE THIS PASSAGE?";
 1410 LINPUT -Z8$
 1420 IF Z8$="N" THEN 1440
 1430 LINPUT "TITLE OF TEXT",-C1$
-1440 PRINT HEX(03)
 1450 LINPUT "DO YOU WANT TO PRINT ONE OR TWO SUGGESTED SUBSTITUTES? (1/2)",-Z
      2$
 1460 PRINT HEX(03)
 1470 Z3$="N"
 1480 LINPUT "IS TEXT IN UPPER CASE LETTERS ONLY?",-Z3$
 1510 B0=999
 1515 J=1
 1520 Y4$="N"
 1521 D1$()=ALL(20)
 1522 Y2$="N"
 1530 Y8$="N"
 1535 Z4$="N"
 1540 H1$="N"
 1550 Z6$="Y"
 1555 Z7$="Y"
 1562 L2$=" "
 1570 S3=0
 1575 S1=0
 1577 P7=1
 1578 L4=99
 1579 U1$()=ALL(20)
 1580 S2=0
 1581 P5=0
 1582 W1=0
 1583 Z5$="Y"
 1584 INIT(FF) U1$()
 1585 INIT(00) V1$(),V2$()
 1610 PRINT HEX(03)
 1620 LINPUT "ENTER THE DEVICE FOR PRINTING THE ANALYSIS RESULTS",-A$
 1630 PRINT HEX(03)
 1640 PRINT AT(12,30);"PRINTING TEXT NOW..."
 1645 SELECT PRINT [A$] (80)
 1649 REM GOTO PHRASE ANALYSIS SUBROUTINE
-1650 GOSUB '100
 1654 REM IF END OF FILE GO TO PRINT ROUTINE
 1655 IF Z4$="Y" THEN 1690
 1669 REM IF SWITCH FOR WORD ANALYSIS [] Y THEN GO TO PRINT ROUTINE
 1670 IF Z7$[]"Y" THEN 1690
 1679 REM GO TO WORD ANALYSIS SUBROUTINE
 1680 GOSUB '210
 1689 REM GO TO PRINT ROUTINE
-1690 GOSUB '240
 1700 IF Z4$="Y" THEN 1275
 1710 GOTO 1650

 1720 REM                                   END OF PROGRAM
-1730 PRINT HEX(03)
 1740 FOR I=1 TO 5
 1750 PRINT AT(10,10);"END OF TEXT ANALYSIS PROGRAM"
 1760 SELECT P3
 1770 NEXT I
 1775 SELECT P0
 1780 LOAD DC T#0,"START5"

 2000 REM %

   PHRASE ANALYSIS SUBROUTINE

 2001 DEFFN '100
 2005 Y3$="Y"
 2006 Y8$="N"
 2009 REM GET FIRST WORD IN A PHRASE
 2010 GOSUB '130
 2019 REM IF END OF FILE THEN RETURN
 2020 IF Z4$="Y" THEN 2590
 2021 REM IF SUPPRESS ANALYSIS = Y THEN RETURN
 2022 IF Y8$="Y" THEN 2590
 2029 REM CHECK FOR CHAPTER HEADING
 2030 IF STR(T1$,1,1)[] "[" THEN 2040
 2031 T3$=T1$
 2032 H1$="Y"
 2034 GOTO 2590

 2039 REM SAVE TEXT POINTER
-2040 B1=B0
 2049 REM INCREMENT WORD COUNT
 2050 W1=W1+1
 2059 REM UPDATE SYLLABLE COUNT
 2060 S2=S2+T2
 2069 REM INCREMENT WORDS PER SENTENCE COUNT
 2070 S3=S3+1
 2109 REM SAVE CURRENT WORD OF TEXT
 2110 T6$=T1$
 2115 T5$=T2$
 2120 T4$=T2$
 2130 T3$=T1$
 2131 REM IF SWITCH FOR PHRASE ANALYSIS = N OR END OF SENTENCE = Y THEN RETURN

 2132 IF Z6$="N" THEN 2490
 2133 IF Z5$="Y" THEN 2490
 2134 REM TURN OFF FIRST WORD IN A PHRASE SWITCH
 2135 Y3$="N"
 2139 REM GET NEXT WORD FOR POSSIBLE PHRASE
-2140 GOSUB '130
 2149 REM IF END OF FILE OR NOT POSSIBLE PHRASE THEN STOP PHRASE ANALYSIS
 2150 IF Z4$="Y" THEN 2330
 2160 IF Z6$="N" THEN 2330
 2169 REM INCREMENT WORDS PER PHRASE AND SYLLABLES PER PHRASE
 2170 T3=T3+1
 2180 T5=T5+T2
 2189 REM IF PHRASE LENGTH ] 30 THEN STOP PHRASE ANALYSIS
 2190 IF LEN(T6$)+LEN(T1$) +1 ] 30 THEN 2330
 2199 REM ADD NEXT WORD TO PARTIAL PHRASE
 2200 STR(T6$,LEN(T6$)+2)=T1$
 2210 STR(T4$,LEN(T4$)+2)=T2$
 2219 REM GO TO CHECK PHRASES TABLE
 2220 GOSUB '160
 2229 REM IF NOT A PARTIAL PHRASE THEN STOP PHRASE ANALYSIS
 2230 IF Y1$="N" THEN 2330
 2259 REM IF NOT A WHOLE PHRASE THEN GO TO ADD NEXT WORD
 2260 IF Y2$="N" THEN 2140
 2269 REM SAVE WHOLE PHRASE AND COUNTERS
 2270 T7$=T6$
 2280 T4=T3
 2290 T6=T5
 2310 B1=B0
 2320 GOTO 2140

 2329 REM IF NO WHOLE PHRASE THEN EXIT ROUTINE
-2330 IF Y2$="N" THEN 2480
 2339 REM UPDATE SYLLABLE, WORDS PER SENTENCE AND WORDS COUNTERS
 2340 S2=S2+T6
 2350 S3=S3+T4
 2360 W1=W1+T4
 2374 REM SET OFF WORD ANALYSIS SWITCH
 2375 Z7$="N"
 2379 REM PUT PHRASE AND SUBSTITUTE(S) IN BRACKETS
 2380 T3$="[" & T7$ & "][*" & STR(P2$(P3),31,25)
 2390 IF Z2$="2" THEN T3$=T3$ & "/" & STR(P2$(P3),56,25)
 2395 T3$=T3$ & "*]"
 2399 REM IF NOT END OF SENTENCE THEN EXIT ROUTINE
 2400 IF Z5$="Y" AND Z4$="N" THEN Z5$="N"
    : IF Y5$="Y" THEN Z5$="Y"
    : IF Z5$[] "Y" THEN 2490
 2409 REM IF NUMBER OF WORDS PER SENTENCE ] 22 THEN PUT WORD COUNT
    : REM ON END OF SENTENCE
 2410 IF S3 ] 22 THEN 2430
 2420 GOTO 2460

-2430 IF S3]99 THEN S3=99
    : CONVERT S3 TO V1$, (##)
 2450 T3$=T3$ & "$$" & V1$ & "$$"
-2460 S3=0
 2475 GOTO 2490

 2479 REM RESET SWITCHES
-2480 Z5$="N"
-2490 T2$=T5$
 2495 T1$=T3$
 2500 T3=0
 2510 T4=0
 2520 T5=0
 2530 T6=0
 2560 Y2$="N"
 2570 Y1$="N"
 2580 B0=B1
 2584 REM IF END OF SENTENCE THEN INCREMENT SENTENCE COUNTER
 2585 IF Z5$="Y" THEN S1=S1+1
-2590 RETURN

 2600 REM %

   GET NEXT WORD SUBROUTINE

 2610 DEFFN '130
 2615 Z6$="Y"
    : Z7$="Y"
 2620 IF B0[200 THEN 2640
 2629 REM GO TO GET NEXT LINE SUBROUTINE
 2630 GOSUB '135
 2639 REM LOOK FOR NON-BLANK CHARACTER IN TEXT
-2640 MAT SEARCH D1$() [B0,M0-B0],[] " " TO B1$()
 2650 IF B1$(1)[] HEX(0000) THEN 2700
 2659 REM IF NO CHARACTERS FOUND THEN GO TO GET NEXT LINE SUBROUTINE
 2660 GOSUB '135
 2669 REM IF NOT THE END OF TEXT GO BACK TO SEARCH FOR NEXT WORD
 2670 IF Y4$[] "Y" THEN 2640
 2678 REM IF END OF TEXT AND FIRST WORD IN A PHRASE THEN SET END OF FILE SWITC
      H ON
 2679 REM IF END OF TEXT AND NOT FIRST WORD IN A PHRASE THEN SET PHRASE ANALYS
      IS SWITCH OFF
 2680 IF Y3$ = "Y" THEN Z4$ = "Y"
    : ELSE Z6$="N"
 2689 REM GO TO RETURN
 2690 GOTO 2960

 2697 REM SKIP OVER BLANK SPACES IN TEXT
 2698 REM IF UP ARROW IN FIRST COLUMN OF TEXT THEN SEND THAT WHOLE
 2699 REM LINE OF TEXT TO PRINT ROUTINE WITHOUT ANALYZING IT
-2700 B0=B0+VAL(B1$(1),2)-1
 2701 IF STR(D1$(),B0,1)[]"!" THEN 2710
 2702 IF Y3$[]"Y" THEN Z6$="N"
    : ELSE GOTO 2704
 2703 GOTO 2960

-2704 IF B0=1 OR B0=81 OR B0=161 THEN 2706
 2705 GOTO 2780

-2706 T3$=D1$(((B0-1)/80)+1)
    : STR(T3$,1,1)=" "
 2707 B0=B0+80
 2708 Z6$="N"
```

```
     : Z7$="N"
     : Y8$="Y"
 2709 GOTO 2960

-2710 IF STR(D1$(),BO,1)[] "[" THEN 2780
 2719 REM CHECK FOR HEADING IN BRACKETS
 2720 MAT SEARCH D1$() [BO,MO-BO], = "]" TO B1$()
 2730 IF B1$(1) = HEX(0000) THEN 2780
 2740 T1$=STR(D1$(),BO,VAL(B1$(1),2))
 2749 REM TURN OFF WORD AND PHRASE ANALYSIS SWITCHES
 2750 Z6$="N"
 2760 Z7$="N"
 2765 BO=BO+VAL(B1$(1),2)
 2769 REM GO TO RETURN
 2770 GOTO 2960

 2779 REM LOOK FOR NEXT SPACE IN TEXT
-2780 MAT SEARCH D1$() [BO,MO-BO], = " " TO B1$()
 2790 IF B1$(1)[] HEX(0000) THEN 2860
 2799 REM IF TEXT POINTER IS NOT AT END OF TEXT THEN CONTINUE
 2800 IF BO[]240 THEN 2830
 2808 REM IF END OF TEXT AND FIRST WORD IN A PHRASE THEN SET END OF FILE SWITC
      H ON
 2809 REM IF END OF TEXT AND NOT FIRST WORD IN A PHRASE THEN SET PHRASE ANALYS
      IS SWITCH OFF
 2810 IF Y3$="Y" THEN Z4$="Y"
     : ELSE Z6$="N"
 2819 REM GO TO RETURN
 2820 GOTO 2960

 2829 REM SAVE WORD OF TEXT AND MOVE POINTER
-2830 T1$=STR(D1$(),BO,MO-BO)
 2840 BO=240
 2850 GOTO 2880

-2860 T1$=STR(D1$(),BO,VAL(B1$(1),2)-1)
 2870 BO=BO+VAL(B1$(1),2)-1
 2879 REM GO TO STRIP PUNCTUATION SUBROUTINE
-2880 GOSUB '140
 2889 REM IF UPPER CASE TEXT AND NOT A NUMBER THEN LEAVE PHRASE ANALYSIS SWITC
      H ON
 2890 IF Z3$="Y" AND STR(T2$,1,1)]HEX(40) THEN 2940
 2899 REM IF LOWER CASE WORD THEN LEAVE PHRASE ANALYSIS SWITCH ON
 2900 IF STR(T2$,1,1) ] HEX(60) THEN 2940
 2909 REM IF FIRST WORD OF A SENTENCE AND NOT A NUMBER THEN LEAVE PHRASE ANALY
      SIS SWITCH ON
 2910 IF Y5$="Y" AND STR(T2$,1,1)]HEX(40) THEN 2940
 2919 REM TURN OFF SWITCH TO PERFORM PHRASE ANALYSIS
 2920 Z6$="N"
 2939 REM CONVERT LOWER CASE LETTERS TO UPPER CASE
-2940 $TRAN(T2$,L1$)R
 2949 REM GO TO COUNT SYLLABLES SUBROUTINE
 2950 GOSUB '150
-2960 RETURN

 3000 REM %

    GET NEXT LINE SUBROUTINE

 3010 DEFFN '135
 3019 REM IF NOT THE FIRST TIME INTO THIS SUBROUTINE THEN GET NEXT LINE
 3020 IF BO[] 999 THEN 3070
    : REM IF TEXT FILENAME IS NOT A FILE OF FILENAMES THEN SKIP NEXT FEW LINES

 3021 IF STR(AO$,5,4)[]"0000" THEN 3028
    : REM OPEN AND LOAD TABLE OF FILENAMES
 3022 DATA LOAD DC OPEN T#6,STR(AO$,1,8)
 3023 DATA LOAD DC #6,F2$()
    : REM INCREMENT INDEX TO FILENAMES TABLE
-3024 K=K+1
    : REM IF FILENAME BEGINS WITH HEX(FFFF) THEN GO TO END
 3025 IF STR(F2$(K),1,2)=HEX(FFFF) THEN 3202
 3026 AO$=F2$(K)
    : REM IF FILENAME IS NO GOOD THEN GET NEXT NAME FROM TABLE
 3027 Y9$="Y"
    : LIMITS T#4,AO$,L1,L2,L3,S
    : IF S[]2 THEN 3024
    : REM OPEN TEXT FILE AND LOAD FIRST THREE LINES OF TEXT
-3028 DATA LOAD DC OPEN T#4,STR(AO$,1,8)
 3029 DATA LOAD DC #4,B$()
    : REM IF FIRST PASSAGE THEN COPY THREE LINES FROM INPUT BUFFER TO WORK ARE
      A
 3030 IF K[]1 AND Y9$="Y" THEN 3180
 3031 MAT COPY B$() TO D1$()
 3039 REM SET TEXT POINTER AND LINE INDEX
 3040 BO=1
 3050 J=4
 3059 REM GO TO RETURN
 3060 GOTO 3210

 3068 REM IF ALL THREE LINES FROM INPUT BUFFER HAVE BEEN LOADED
 3069 REM TO THE WORK AREA THEN GO READ THREE MORE LINES FROM DISK
-3070 IF J=4 THEN 3150
 3079 REM SKIP OVER BLANK LINES
```

```
-3080 IF B$(J)=" " THEN J=J+1
    : ELSE GOTO 3100
 3090 GOTO 3070

 3099 REM MOVE WORK AREA TEXT UP ONE LINE AND ADD INPUT BUFFER LINE
-3100 D1$(1)=D1$(2)
 3110 D1$(2)=D1$(3)
 3120 D1$(3)=B$(J)
 3130 J=J+1
 3131 REM RE-POSITION TEXT POINTERS
 3132 IF BO[81 THEN 3210
 3135 BO=BO-80
    : B1=B1-80
 3139 REM GO TO RETURN
 3140 GOTO 3210

 3149 REM IF NOT END OF PASSAGE THEN READ THREE MORE LINES FROM DISK
-3150 IF Y4$="Y" THEN 3210
 3160 DATA LOAD DC #4,B$()
 3170 IF END THEN 3200
-3180 J=1
 3190 GOTO 3080

 3199 REM IF MULTIPLE PASSAGES THEN GO GET NEXT PASSAGE
-3200 IF Y9$="Y" THEN 3024
-3202 Y4$="Y"
-3210 RETURN

 3220 REM %

      STRIP PUNCTUATION SUBROUTINE

 3230 DEFFN '140
 3240 T2$=T1$
 3249 REM IF LAST WORD WAS END OF SENTENCE THEN THIS WORD IS THE FIRST WORD OF
       A SENTENCE
 3250 IF Z5$="Y" THEN Y5$="Y"
    : ELSE Y5$="N"
 3260 Z5$="N"
 3269 REM SEARCH FOR PUNCTUATION AT THE BEGINNING OF THE WORD
-3270 MAT SEARCH P6$(), =STR(T2$,1,1) TO B1$()
 3280 IF B1$(1)=HEX(0000) THEN 3320
 3289 REM TURN OFF PHRASE ANALYSIS SWITCH AND TAKE OFF PUNCTUATION
 3290 Z6$="N"
 3300 T2$=STR(T2$,2)
 3310 GOTO 3270
 3319 REM SEARCH FOR END OF SENTENCE PUNCTUATION
-3320 MAT SEARCH P5$(), =STR(T2$,LEN(T2$),1) TO B1$()
 3329 REM IF NO PUNCTUATION THEN SKIP TO 3360
 3330 IF B1$(1)=HEX(0000) THEN 3360
 3334 REM IF FIRST WORD IN A PHRASE THEN DON'T DO PHRASE ANALYSIS
 3335 IF Y3$="Y" THEN Z6$="N"
 3339 REM TURN ON END OF SENTENCE SWITCH
 3340 Z5$="Y"
 3344 REM TAKE OFF PUNCTUATION MARK
 3345 IF LEN(T2$)=1 THEN 3440
 3350 T2$=STR(T2$,1,LEN(T2$)-1)
 3359 REM SEARCH FOR OTHER PUNCTUATION ON END OF WORD
-3360 MAT SEARCH P6$(), =STR(T2$,LEN(T2$),1) TO B1$()
 3370 IF B1$(1)=HEX(0000) THEN 3400
 3374 REM IF FIRST WORD IN A PHRASE THEN DON'T DO PHRASE ANALYSIS
 3375 IF Y3$="Y" THEN Z6$="N"
 3376 REM TAKE OFF PUNCTUATION MARK
 3377 IF LEN(T2$)=1 THEN 3440
 3380 T2$=STR(T2$,1,LEN(T2$)-1)
 3389 REM GO BACK AND CHECK FOR ANY MORE PUNCTUATION
 3390 GOTO 3320

 3395 REM CHECK FOR POSSIBLE ABBREVIATION AND TURN OFF END OF SENTENCE SWITCH
      IF IT IS AN ABBREVIATION
-3400 IF STR(T1$,LEN(T1$),1)[]"." THEN 3430
 3410 MAT SEARCH X2$(), =STR(T1$,1,LEN(T1$)-1) TO B1$()
 3420 IF B1$(1)[]HEX(0000) THEN Z5$="N"
-3430 IF LEN(T2$) [ 3 THEN 3440
 3434 REM TAKE OFF 'S
 3435 IF STR(T2$,LEN(T2$)-1,2)= "'S" OR STR(T2$,LEN(T2$)-1,2)="'s" THEN T2$=ST
      R(T2$,1,LEN(T2$)-2)
-3440 RETURN

 3450 REM %

      COUNT SYLLABLES SUBROUTINE

 3460 DEFFN '150
 3464 REM IF WORD IS A NUMBER OR [ 4 CHARACTERS THEN IT IS ONE SYLLABLE
 3465 IF STR(T2$,1,1) [ HEX(41) THEN 3480
 3470 IF LEN(T2$) ] 3 THEN 3500
-3480 T2=1
 3489 REM GO TO RETURN
 3490 GOTO 3610

-3500 T9$=T2$
 3510 T8$()=ALL(20)
 3519 REM TAKE OFF ED AND ES ENDINGS
 3520 IF STR(T9$,LEN(T9$)-1,2)="ED" THEN 3540
 3530 IF STR(T9$,LEN(T9$)-1,2)[]"ES" THEN 3560
-3540 T9$=STR(T9$,1,LEN(T9$)-2)
 3550 GOTO 3580

 3559 REM TAKE OFF E ENDINGS BUT LEAVE ON LE ENDING
-3560 IF STR(T9$,LEN(T9$)-1,2)="LE" THEN 3580
 3570 IF STR(T9$,LEN(T9$),1)="E" THEN T9$=STR(T9$,1,LEN(T9$)-1)
 3579 REM CHANGE ALL CONSONANTS TO SPACES
-3580 $TRAN(T9$,C2$)R
 3589 REM COPY EACH VOWEL OR VOWEL STRING INTO ITS OWN ELEMENT IN T8$()
 3590 $UNPACK (D=HEX(0320))T9$ TO T8$()
 3598 REM CALCULATE THE NUMBER OF ELEMENTS WITH VOWELS IN THEM
 3599 REM THIS VALUE = THE NUMBER OF SYLLABLES
 3600 T2=INT(LEN(T8$())/4)+1
-3610 RETURN

 3620 REM %

    CHECK PHRASES TABLE SUBROUTINE

 3630 DEFFN '160
 3639 REM SEARCH TABLE OF PHRASES
 3640 MAT SEARCH P2$(), =T4$ TO B1$() STEP 80
 3650 IF B1$(1) = HEX(0000) THEN 3720
 3651 REM CALCULATE ELEMENT NUMBER OF MATCHING PHRASE IN THE TABLE
 3652 P3=((VAL(B1$(1),2)-1)/80)+1
 3655 P1$=STR(P2$(P3),1,30)
 3660 IF LEN(P1$) = LEN(T4$) THEN 3690
 3669 REM SET PARTIAL PHRASE ON, WHOLE PHRASE OFF
 3670 Y1$="Y"
 3675 Y2$="N"
 3680 GOTO 3730

 3689 REM SET PARTIAL AND WHOLE PHRASE ON
-3690 Y2$="Y"
 3700 Y1$="Y"
 3710 GOTO 3730

 3719 REM SET PARTIAL PHRASE OFF
-3720 Y1$="N"
-3730 RETURN

 3800 REM %

   WORD ANALYSIS SUBROUTINE
 3810 DEFFN '210
 3811 REM IF WORD LENGTH = 1 OR THE WORD IS A NUMBER DON'T ANALYZE IT
 3812 IF LEN(T2$)=1 THEN 4000
 3815 T1$=T3$
 3845 IF STR(T2$,1,1)[HEX(41) THEN 4000
    : REM IF NOT UPPER CASE TEXT AND NOT FIRST WORD OF A SENTENCE AND FIRST LE
      TTER IS A CAPITAL THEN DON'T ANALYZE
 3846 IF Z3$="N" AND Y5$="N" AND STR(T1$,1,1)[HEX(61) THEN 4000
 3849 REM SEARCH MOST COMMON WORDS TABLE
 3850 MAT SEARCH X1$(), =STR(T2$,1,10) TO B1$() STEP 10
 3860 IF B1$(1)[]HEX(0000) THEN 4000
 3869 REM GO TO SEARCH WORD SUBSTITUTION SUBROUTINE
 3870 GOSUB '230
 3879 REM IF NO MATCH CONTINUE ANALYSIS
 3880 IF Y7$="N" THEN 3904
 3884 REM PUT WORD AND SUBSTITUTE(S) IF ANY IN BRACKETS
 3885 IF STR(W5$(P4),26,20)=" " THEN 4000
 3890 T3$="[" & T1$ & "][*" & STR(W5$(P4),26,20)
 3900 IF Z2$="2" THEN T3$=T3$ & "/" & STR(W5$(P4),46,15)
 3901 T3$=T3$ & "*]"
 3902 GOTO 4000

 3903 REM GO TO CHECK SUPPLEMENTARY WORD LIST SUBROUTINE
-3904 IF Z1$="N" THEN 3911
 3905 GOSUB '220
 3906 IF Y6$="Y" THEN 4000
 3907 REM STRIP PUNCTUATION FROM LOWER CASE TEXT WORD
-3911 MAT SEARCH P5$(), =STR(T1$,LEN(T1$),1) TO B1$()
 3912 IF B1$(1) [] HEX(0000) THEN 3916
 3913 MAT SEARCH P6$(), =STR(T1$,LEN(T1$),1) TO B1$()
 3914 IF B1$(1) [] HEX(0000) THEN 3916
 3915 GOTO 3918

-3916 T1$ = STR(T1$,1,LEN(T1$)-1)
 3917 GOTO 3911

-3918 MAT SEARCH P6$(), =STR(T1$,1,1) TO B1$()
 3919 IF B1$(1) [] HEX(0000) THEN T1$ = STR(T1$,2)
    : REM SEARCH UNCOMMON WORDS TABLE
 3920 MAT SEARCH U1$(), =T1$ TO B1$() STEP 21
 3930 IF B1$(1)=HEX(0000) THEN 3960
 3939 REM ADD ONE TO FREQUENCY COUNT OF UNCOMMON WORD IN TABLE
 3940 STR(U1$(INT(VAL(B1$(1),2)/21)+1),21,1)= ADD HEX(01)
 3950 GOTO 3990

 3959 REM ADD WORD TO UNCOMMON WORDS TABLE
-3960 P5=P5+1
 3970 U1$(P5)=T1$
 3980 STR(U1$(P5),21,1)= HEX(01)
-3990 T1$=T3$
 3994 REM PUT BRACKETS AROUND UNCOMMON WORD
 3995 T3$ = "[" & T1$ & "]"
 3999 REM IF NOT END OF SENTENCE THEN RETURN
-4000 IF Z5$="N" THEN 4080
 4010 IF S3 ] 22 THEN 4030
 4020 GOTO 4060

 4029 REM PUT WORDS PER SENTENCE AT END OF SENTENCE
-4030 IF S3]99 THEN S3=99
    : CONVERT S3 TO V1$,(##)
 4050 T3$=T3$ & "$$" & V1$ & "$$"
-4060 S3=0
-4080 RETURN

 4100 REM %

     CHECK SUPPLEMENTARY WORD LISTS

 4110 DEFFN '220
 4115 Y6$="N"
 4229 REM SEARCH BLOCK POINTER TABLE FOR POINTER ] WORD
 4230 MAT SEARCH W4$(), ]STR(T2$,1,3) TO B1$() STEP 3
 4239 REM CALCULATE BLOCK NUMBER
 4240 P6=INT(VAL(B1$(1),2)/3)
 4245 IF P6=1 THEN 4320
 4249 REM LOAD BLOCK OF WORDS FROM DISK
-4250 DATA LOAD DA T#5, (U7+(P6-2)*9) W3$()
 4259 REM SEARCH BLOCK OF WORDS FOR MATCH WITH TEXT WORD
 4260 MAT SEARCH W3$(), =STR(T2$) TO B1$() STEP 20
 4270 IF B1$(1)[]HEX(0000) THEN 4310
 4278 REM IF FIRST THREE LETTERS OF TEXT WORD = FIRST THREE
 4279 REM LETTERS OF FIRST WORD IN BLOCK THEN GO BACK AND SEARCH PREVIOUS BLOC
      K
 4280 IF STR(T2$,1,3)[] STR(W3$(1),1,3) THEN 4320
 4290 P6=P6-1
 4300 GOTO 4250

 4309 REM TURN ON SWITCH FOR FOUND WORD ON LIST
-4310 Y6$="Y"
-4320 RETURN

 4330 REM %
```

CHECK WORD SUBSTITUTION LIST

```
 4340 DEFFN '230
 4345 Y7$="N"
 4349 REM SEARCH BLOCK POINTER TABLE FOR POINTER ] WORD
 4350 MAT SEARCH W6$(), ]STR(T2$,1,3) TO B1$() STEP 3
 4359 REM CALCULATE BLOCK NUMBER
 4560 P6=INT(VAL(B1$(1),2)/3)
 4569 REM LOAD BLOCK OF WORDS FROM DISK
-4570 DATA LOAD DA T#2, (U4+(P6-1)*50) W5$()
 4579 REM SEARCH BLOCK OF WORDS FOR MATCH WITH TEXT WORD
 4580 MAT SEARCH W5$(), =STR(T2$,1,20) TO B1$() STEP 60
 4590 IF B1$(1)=HEX(0000) THEN 4630
 4599 REM CALCULATE ELEMENT NUMBER OF MATCHING WORD IN TABLE
 4600 P4=INT(VAL(B1$(1),2)/60)+1
 4609 REM SET FOUND MATCHING WORD SWITCH ON
 4610 Y7$="Y"
 4620 GOTO 4660

 4628 REM IF FIRST THREE LETTERS OF TEXT WORD = FIRST THREE LETTERS
 4629 REM OF FIRST WORD IN BLOCK THEN GO BACK AND SEARCH PREVIOUS BLOCK
-4630 IF STR(T2$,1,3)[] STR(W5$(1),1,3) THEN 4660
 4635 IF P6=1 THEN 4660
 4640 P6=P6-1
 4650 GOTO 4570

-4660 RETURN

 4700 REM %
```

PRINT SUBROUTINE

```
 4710 DEFFN '240
    : REM PRINT A LINE IF ANALYSIS SUPPRESS SWITCH IS ON
 4711 IF Y8$="N" THEN 4720
 4712 IF L2$[]" " THEN GOSUB '242
 4713 L2$=T3$
 4714 GOSUB '242
    : L2$=" "
 4715 GOTO 4860

 4719 REM CHECK IF END OF FILE
-4720 IF Z4$="N" THEN 4750
 4724 REM IF LINE BUFFER IS NOT BLANK THEN PRINT IT
 4725 IF L2$[]" " THEN GOSUB '242
 4729 REM GO TO PRINT READABILITY SUBROUTINE
 4730 GOSUB '244
 4739 REM GO TO RETURN
 4740 GOTO 4860

 4749 REM IF HEADING SWITCH IS ON THEN PRINT HEADING
-4750 IF H1$="Y" AND L2$=" " THEN 4780
 4760 IF H1$[]"Y" THEN 4810
 4765 H1$="N"
 4770 GOSUB '242
 4775 H1$="Y"
-4780 L2$=T3$
 4790 GOSUB '242
 4795 L2$=" "
 4800 GOTO 4860

 4809 REM IF CURRENT LINE + NEXT WORD ] 80 THEN PRINT CURRENT LINE
-4810 IF LEN(L2$) + LEN(T3$) + 1 [ 81 THEN 4850
 4820 GOSUB '242
 4830 L2$=T3$
 4840 GOTO 4860

 4849 REM IF CURRENT LINE + NEXT WORD [ 81 THEN ADD WORD TO LINE BUFFER
-4850 STR(L2$,LEN(L2$)+2)=T3$
-4860 RETURN

 4900 REM %

    PRINT A LINE SUBROUTINE

 4910 DEFFN '242
 4919 REM IF HEADING SWITCH IS OFF SKIP TO LINE 5000
 4920 IF H1$[]"Y" THEN 5000
 4929 REM PAGE COUNT = 1
 4930 P7=1
 4939 REM SKIP TO TOP OF FORM AND PRINT PAGE NUMBER
 4940 PRINT HEX(0C)
 4950 PRINT TAB(70);"PAGE  ";P7
 4951 REM CENTER AND PRINT TITLE IF THERE IS ONE
 4952 IF Z8$="Y" THEN PRINT TAB(INT((80-LEN(C1$)*2)/4));HEX(0E);C1$
    : ELSE GOTO 4960
 4954 PRINT
 4956 Z8$="N"
 4959 REM CENTER AND PRINT HEADING
-4960 PRINT TAB(INT((80-(LEN(L2$)-2)*2)/4));HEX(0E);STR(L2$,2,LEN(L2$)-2)
 4970 PRINT
 4975 H1$="N"
 4979 REM LINE COUNT = 4 AND PAGE NUMBER = 2
 4980 L4=4
 4985 P7=2
 4990 GOTO 5090

 4999 REM IF PAGE NOT FULL SKIP TO LINE 5060
-5000 IF L4[55 THEN 5060
 5009 REM SKIP TO TOP OF FORM AND PRINT PAGE NUMBER
 5010 PRINT HEX(OC)
 5020 PRINT TAB(70);"PAGE  ";P7
 5030 PRINT
 5040 L4=3
 5050 P7=P7+1
 5051 REM CENTER AND PRINT TITLE IF THERE IS ONE
 5052 IF Z8$="Y" THEN PRINT TAB(INT((80-LEN(C1$)*2)/4));HEX(OE);C1$
    : ELSE GOTO 5060
 5054 PRINT
 5056 Z8$="N"
 5058 L4=L4+2
 5059 REM PRINT A LINE OF TEXT
-5060 PRINT L2$
 5070 PRINT
 5080 L4=L4+2
-5090 RETURN

 5100 REM %

   PRINT READABILITY SUBROUTINE

 5110 DEFFN '244
 5119 REM SKIP TO TOP OF FORM
 5120 PRINT HEX(OC)
 5129 REM CALCULATE AVERAGE NUMBER OF WORDS PER SENTENCE
 5130 IF S1[]0 THEN F3=W1/S1
    : ELSE F3=0
 5139 REM CALCULATE AVERAGE NUMBER OF SYLLABLES PER WORD
 5140 IF W1[]0 THEN F4=S2/W1
    : ELSE F4=0
 5149 REM CALCULATE GRADE LEVEL
 5150 F1=(F3*.39)+(F4*11.8)-15.59
 5159 REM PRINT READABILITY RESULTS
 5160 PRINTUSING 5450
 5170 PRINT
 5180 PRINTUSING 5460
 5190 PRINTUSING 5470,S1,W1,S2
 5200 PRINT
 5210 PRINTUSING 5480
 5220 PRINTUSING 5490,F3,F4
 5230 PRINT
 5240 PRINTUSING 5500
 5250 PRINTUSING 5510,F1
 5260 PRINT
 5270 PRINTUSING 5520
 5280 PRINT
 5290 PRINTUSING 5530
 5300 PRINT
 5309 REM DEFINE FUNCTION FOR FREQUENCY OF UNCOMMON WORD
 5310 DEFFN F (F)=VAL(STR(U1$(F),21,1))
 5319 REM SORT UNCOMMON WORDS TABLE AND PRINT IT
 5320 MAT SORT U1$() TO V1$(),V2$()
 5330 FOR I=1 TO P5 STEP 2
 5340 V3=VAL(V2$(I+1),2)
 5350 IF STR(U1$(V3),1,1)[]HEX(FF) THEN 5370
 5360 U1$(V3)=" "
-5370 V4=VAL(V2$(I),2)
 5380 IF STR(U1$(V3),1,20)=" " THEN 5400
 5390 GOTO 5420

-5400 PRINTUSING 5550,STR(U1$(V4),1,20),FNF(V4)
 5410 GOTO 5430

-5420 PRINTUSING 5540,STR(U1$(V4),1,20),FNF(V4),STR(U1$(V3),1,20),FNF(V3)
-5430 NEXT I
 5440 RETURN

-5450 % ------------------      READABILITY RESULTS      ---------------------
      -
-5460 % Number of Sentences       Number of Words        Number of Syllables
-5470 %     ######                  ######                  ######
-5480 % Avg. Number of Words per Sentence     Avg. Number of Syllables per Wor
      d
-5490 %             #####.##                              #####.##
-5500 %  GRADE LEVEL   (Based on DOD Readability Standard)
-5510 %      ####.#
-5520 % ------------------ WORDS NOT ON COMMON WORD LISTS --------------------
      -
-5530 % WORD                  FREQ             WORD                  FREQ
-5540 % ####################   ##              ####################   ##
-5550 % ####################   ##
 6999 REM TABLE OF MOST COMMONLY USED WORDS
-7000 X1$(1)="A"
 7002 X1$(2)="ABOUT"
 7004 X1$(3)="AFTER"
 7006 X1$(4)="AGAINST"
 7008 X1$(5)="AGENTS"
 7010 X1$(6)="AIR"
```

```
7012 X1$(7)="AIRCRAFT"
7014 X1$(8)="ALL"
7016 X1$(9)="ALSO"
7018 X1$(10)="AN"
7020 X1$(11)="AND"
7022 X1$(12)="ANY"
7024 X1$(13)="ARE"
7026 X1$(14)="ARMED"
7028 X1$(15)="AS"
7030 X1$(16)="AT"
7032 X1$(17)="BACK"
7034 X1$(18)="BE"
7036 X1$(19)="BECAUSE"
7038 X1$(20)="BEEN"
7040 X1$(21)="BEFORE"
7042 X1$(22)="BEING"
7044 X1$(23)="BOTH"
7046 X1$(24)="BUT"
7048 X1$(25)="BY"
7050 X1$(26)="CAN"
7052 X1$(27)="CLASS"
7054 X1$(28)="COURT"
7056 X1$(29)="CRAFT"
7058 X1$(30)="DISCHARGE"
7060 X1$(31)="DO"
7062 X1$(32)="DURING"
7064 X1$(33)="DUTY"
7066 X1$(34)="EACH"
7068 X1$(35)="END"
7070 X1$(36)="ENLISTED"
7072 X1$(37)="EQUIPMENT"
7074 X1$(38)="FIG"
7076 X1$(39)="FIRE"
7078 X1$(40)="FIRST"
7080 X1$(41)="FOR"
7082 X1$(42)="FORCES"
7084 X1$(43)="FROM"
7086 X1$(44)="GENERAL"
7088 X1$(45)="HAND"
7090 X1$(46)="HAS"
7092 X1$(47)="HAVE"
7094 X1$(48)="HE"
7096 X1$(49)="HIM"
7098 X1$(50)="HIS"
7100 X1$(51)="HOWEVER"
7102 X1$(52)="IF"
7104 X1$(53)="IN"
```

```
7106 X1$(54)="INTO"
7108 X1$(55)="IS"
7110 X1$(56)="IT"
7112 X1$(57)="ITS"
7114 X1$(58)="LEAVE"
7116 X1$(59)="LINE"
7118 X1$(60)="MAKE"
7120 X1$(61)="MAN"
7122 X1$(62)="MANY"
7124 X1$(63)="MASK"
7126 X1$(64)="MAY"
7128 X1$(65)="MEANS"
7130 X1$(66)="MEN"
7132 X1$(67)="MILITARY"
7134 X1$(68)="MORE"
7136 X1$(69)="MOST"
7138 X1$(70)="MUST"
7140 X1$(71)="NAVAL"
7142 X1$(72)="NAVY"
7144 X1$(73)="NO"
7146 X1$(74)="NOT"
7148 X1$(75)="NUCLEAR"
7150 X1$(76)="OF"
7152 X1$(77)="OFFICER"
7154 X1$(78)="OFFICERS"
7156 X1$(79)="ON"
7158 X1$(80)="ONE"
7160 X1$(81)="ONLY"
7162 X1$(82)="OR"
7164 X1$(83)="ORDER"
7166 X1$(84)="OTHER"
7168 X1$(85)="OUR"
7170 X1$(86)="OUT"
7172 X1$(87)="OVER"
7174 X1$(88)="PAY"
7176 X1$(89)="PERSON"
7178 X1$(90)="PERSONNEL"
7180 X1$(91)="PETTY"
7182 X1$(92)="PLACE"
7184 X1$(93)="REQUIRED"
7186 X1$(94)="RIFLE"
7188 X1$(95)="RIGHT"
7190 X1$(96)="SALUTE"
7192 X1$(97)="SAME"
7194 X1$(98)="SERVICE"
7196 X1$(99)="SEVERAL"
7198 X1$(100)="SHIP"
```

```
7200 X1$(101)="SHIPS"
7202 X1$(102)="SHOULD"
7204 X1$(103)="SIDE"
7206 X1$(104)="SO"
7208 X1$(105)="SOME"
7210 X1$(106)="SPECIAL"
7212 X1$(107)="STATES"
7214 X1$(108)="SUCH"
7216 X1$(109)="SYSTEM"
7218 X1$(110)="TAKE"
7220 X1$(111)="THAN"
7222 X1$(112)="THAT"
7224 X1$(113)="THE"
7226 X1$(114)="THEIR"
7228 X1$(115)="THEM"
7230 X1$(116)="THEN"
7232 X1$(117)="THERE"
7234 X1$(118)="THESE"
7236 X1$(119)="THEY"
7238 X1$(120)="THIS"
7240 X1$(121)="THROUGH"
7242 X1$(122)="TIME"
7244 X1$(123)="TO"
7246 X1$(124)="TRAINING"
7248 X1$(125)="TWO"
7250 X1$(126)="TYPE"
7252 X1$(127)="TYPES"
7254 X1$(128)="UNDER"
7256 X1$(129)="UNIFORM"
7258 X1$(130)="UNITED"
7260 X1$(131)="UP"
7262 X1$(132)="USE"
7264 X1$(133)="USED"
7266 X1$(134)="USUALLY"
7268 X1$(135)="WAR"
7270 X1$(136)="WAS"
7272 X1$(137)="WATER"
7274 X1$(138)="WERE"
7276 X1$(139)="WHEN"
7278 X1$(140)="WHERE"
7280 X1$(141)="WHICH"
7282 X1$(142)="WHO"
7284 X1$(143)="WILL"
7286 X1$(144)="WITH"
7288 X1$(145)="WITHOUT"
7290 X1$(146)="WORN"
7292 X1$(147)="YOU"
```

```
7294 X1$(148)="YOUR"
7296 REM THIS IS A TABLE OF ABBREVIATIONS DEVELOPED BY BELL LABS
7298 X2$(1)="Ch"
   : X2$(2)="ch"
   : X2$(3)="ckts"
7300 X2$(4)="dB"
   : X2$(5)="Dept"
   : X2$(6)="dept"
7310 X2$(7)="Depts"
   : X2$(8)="depts"
   : X2$(9)="Dr"
7320 X2$(10)="Drs"
   : X2$(11)="Eq"
   : X2$(12)="eq"
7330 X2$(13)="etc"
   : X2$(14)="Fig"
   : X2$(15)="fig"
7340 X2$(16)="Figs"
   : X2$(17)="figs"
   : X2$(18)="ft"
7350 X2$(19)="in"
   : X2$(20)="Inc"
   : X2$(21)="Jr"
7360 X2$(22)="jr"
   : X2$(23)="mi"
   : X2$(24)="Mr"
7370 X2$(25)="Mrs"
   : X2$(26)="Ms"
   : X2$(27)="No"
7380 X2$(28)="no"
   : X2$(29)="Nos"
   : X2$(30)="nos"
7390 X2$(31)="Ref"
   : X2$(32)="ref"
   : X2$(33)="Refs"
7400 X2$(34)="refs"
   : X2$(35)="St"
   : X2$(36)="vs"
7410 X2$(37)="yr"
7420 RETURN
```

APPENDIX G

WANG BASIC-2 COMMANDS

The following commands are frequently employed in the computer programs in the appendices. An installation that converts from a WANG BASIC-2 to another version of BASIC will find the list useful.

WANG BASIC-2 COMMANDS

ADD - Used to perform binary addition on alpha-variables.

For example:

STR(U1$(I),21,1)= ADD HEX(01)

This instruction adds 1 to the 21'st byte of U1$(I)

ALL() - Used on the right side of an equal sign, this statement initializes the alphanumeric variable on the left side of the equal sign.

Example: T8$()=ALL(20) initializes T8$() to all hex 20 which are spaces

& - Used to concatenate two or more alpha strings.

Example: If T7$="WANG" then after the statement T8$= "[" & T7$ & "]" is executed T8$="[WANG]"

CONVERT - Converts numeric variables to alpha variables and vice-versa.

Example: CONVERT S3 TO V1$, (##)

This statement converts the numeric value of S3 to an ASCII character string according to the format image (##) and places it in V1$

DATA LOAD DA - Reads from a disk using direct addressing.

Example: DATA LOAD DA T#5, (U7+(P6-2)*9) W3$()

This statement reads from the disk platter selected as #5. The read starts at sector number (U7+(P6-2)*9) and reads 1 or more logical records until W3$() is full.

DATA LOAD DC - Reads from a disk using catalog addressing.

Example: DATA LOAD DC #4,B$()

This statement reads from the file opened as #4. The read starts at the beginning of a logical record and reads enough data to fill B$().

DATA LOAD DC OPEN - Opens a file and assigns a device number to that file.

Example: DATA LOAD DC OPEN T#1,"WORDSUB1"

This statement opens the file with filename "WORDSUB1". It is assigned the file number #1 and must be on the disk selected as #1.

DATA SAVE DC - Writes one logical record to a disk file.

Example: `DATA SAVE DC #1,B$()`

This statement writes the contents of B$() to the file assigned to #1. B$() becomes one logical record on the file.

DATA SAVE DC OPEN - Reserves area on a disk, enters a filename into the disk catalog, and assigns a number to that filename.

Example: `DATA SAVE DC OPEN T$#2,(4000),"TEMPWORK"`

This statement reserves 4000 sectors on the disk platter selected as #2. The filename "TEMPWORK" is entered into the disk catalog for disk T#2 and #2 is the file number which identifies "TEMPWORK". The "$" parameter specifies that a "read-after-write" be performed to verify that the information entered into the disk catalog was written correctly.

DBACKSPACE - Backspaces a disk file.

Example: `DBACKSPACE #1,1` backspaces file #1 one logical record.

`DBACKSPACE #1,BEG` backspaces file #1 to the beginning of the file.

DEFFN' - Identifies the beginning of a subroutine called with a GOSUB' statement

Example: `DEFFN '210 (A$,B,C)`

This statement defines the beginning of subroutine 210 where (A$,B,C) are the parameters passed with a GOSUB '210 (T$,0,1).

DEFFN - Defines a function of one variable.

Example: `DEFFN F (F)=VAL(STR(U1$(F),21,1))`

This statement defines a function called F which is calculated as VAL(STR(U1$(F),21,1)) each time it is called with an FN statement.

DIM - Defines the size and type of a variable.

A variable can be numeric or alpha. A variable is defined as alpha if it has a dollar sign in the name. For alpha variables the length can be assigned with a DIM statement. The default length of alpha variables is 16 bytes. All numeric variables are 8 bytes.

Examples:

A1$8 = An alpha string 8 bytes long

B1$(3)10 = An alpha array with three elements each of which is 10 bytes

C2$(6,2)20 = A 2-dimensional array with 6 rows and 2 columns with each element 20 characters long.

A = A numeric scalar

A1(3) = A numeric array with 3 elements

Note: A1$ and A1$() are two different variables.

When an array is referenced without an index, such as A1$(), then it is treated as one continuous character string.

FN - Used to call a user defined function.

Example:

```
:30 C=3
:40 DEFFN A(X)=X!2-X
:80 PRINT FNA(C*2)
:RUN
30
```

The value in parenthesis (C*2) is evaluated. The resultant value is then used to replace the dummy variable (X) in the DEFFN statement. FNA is then calculated using that value (6↑2-6)=30.

GOSUB - Transfers control to a specified line number.

Example: `GOSUB 7000`

This statement causes the program to branch to line number 7000.

GOSUB' - Transfers control to a specified subroutine.

Example: `GOSUB '210 (A1$,1,0)`

This statement causes the program to branch to the next statement after the DEFFN '210 statement. The parameters (A1$,1,0) are passed to the subroutine.

$GIO - A customized I/O command. This command is not readily translated. The only place it is used in this system is in the text editor. It is suggested that if the user wishes to convert this system to another machine or language then he should use a different text editor.

$IF ON - Used to test whether a device is ready or not-ready.

Example: $IF ON/0007134 causes a program transfer to line 7134 if device 000 is ready.

% image - A formating statement used with PRINTUSING statements. The character # is used as a replacement character.

INIT - Initializes a variable to the hex characters in parenthesis.

Example: INIT(FF) U1$() initializes U1$() to all hex FF's.

INPUT - Allows input from terminal keyboard. Program stops until the carriage return key is pressed.

INT - Integer function returns the largest integer that is less than or equal to the argument.

Example: If X=2.3 then INT(X)=2

If X=2.8 then INT(X)=2

KEYIN - Receives a single character from an input device.

Example: KEYIN X3$(1),7134,7136

This statement receives a single character into X3$(1). If the character is a standard character then the program branches to the first line number. If the character is a special function key then the program branches to the second line number.

LEN - Returns the length in bytes of a character string. Imbedded spaces are counted as one character each. Trailing spaces are not counted.

Example: If T6$="A TEST" then LEN(T6$)=6 even if T6$ is dimensioned with a length > 6.

LIMITS - Places the starting, ending, used (or current) sector addresses of the cataloged disk file in the specified variables. It also determines the status of the disk file.

Form 1 of LIMITS:

```
LIMITS T#1,"TEMPWORK",U1,U2,U3,S
```

With the filename specified ("TEMPWORK"), U1= starting sector, U2= ending sector, U3= number of used sectors, and S= status of file where S=2 for active data file, S=1 for active program file, S=0 for filename not in index, S=-1 for scratched program file, and S=-2 for scratched data file.

Form 2 of LIMITS:

```
LIMITS T#1,U1,U2,U3
```

Without the filename specified, U1= starting sector, U2= ending sector, and U3= current sector address.

LINPUT - Allows input from terminal keyboard. The system is placed in edit mode and the existing contents of the variable are displayed. The operator can then enter data or edit the existing data. The (-) parameter causes underlining for the dimensioned length of the variable.

LOAD - Causes the current program to be cleared from memory and loads and starts execution of the specified program.

Example: `LOAD DC T#0,"START1"`

This statement loads and begins execution of "START1"

MAT COPY - Copies one array to another.

Example: MAT COPY B$() TO D1$() copies the contents of B$() into D1$().

MAT MOVE - Copies the elements from one array to another in the order specified by a third array.

Example: `MAT MOVE F2$(), F4$() TO F5$()`

This instruction copies the elements of F2$() into F5$(). The elements of F4$() are used as the index to F2$(). That is,

if F2$(1)=C	and F4$(1)=3	then F5$(1)=A
F2$(2)=D	F4$(2)=4	F5$(2)=B
F2$(3)=A	F4$(3)=1	F5$(3)=C
F2$(4)=B	F4$(4)=2	F5$(4)=D

This instruction is used in conjunction with the MAT SORT instruction and F4$() would have been the locator variable in a MAT SORT instruction.

MAT SEARCH - This statement scans the search variable for substrings that satisfy the given relation to the specified alpha-variable or literal. The positions of substrings which satisfy the given relation are placed into the locator variable in the order in which they are found; each position is stored as a two-byte binary value specifying the position of the first character of the substring relative to the starting character of the search. If there is any space left in the locator variable when the search ends then the next two bytes are set equal to HEX(0000). If no substrings satisfy the relation then the first two bytes of the locator variable are set to HEX(0000). The step parameter causes the search to start at the first position, the (1+s) position, the (1+2s) position, and so on; where s is the value in the step parameter.

Example:

```
MAT SEARCH D1$() <B0,M0-B0>, = " " TO B1$()
```

This instruction searches D1$() for spaces from location B0 for a length of M0-B0. The position of each space encountered is placed in B1$(). If D1$()= "THIS IS A TEST" and B0=1 then B1$(1)=HEX(05), B1$(2)=HEX(08), and B1$(3)=HEX(0A). The search ends when either the locator variable is full or when the number of characters to be checked in the search variable is less than the length of the alpha-variable or literal in the relation.

MAT SORT - This statement sorts an array by placing 2-byte binary values that correspond to the index values of the array into a locator array. A third array is used as a work area.

Example:

```
  MAT SORT F2$() TO F3$(), F4$()
If F2$(1)=C  then  F4$(1)=3
   F2$(2)=D        F4$(2)=4
   F2$(3)=A        F4$(3)=1
   F2$(4)=B        F4$(4)=2
```

Where F4$() is the locator array and F3$() is the work area.

ON/GOTO - This statement causes a branch to the line number in the position after the word GOTO that corresponds to the value of the numeric variable.

Example: `ON P GOTO 631,632,633`

If P=1 then the program branches to 631.

If P=2 then the program branches to 632.

If P=3 then the program branches to 633.

If P does not equal 1,2, or 3 then the next statement after the ON/GOTO statement is executed.

POS - Returns the position which satisfies the relation in parenthesis.

Example: `POS("YyNn" = STR(Z6$,1,1))`

The statement returns the position of the literal that is equal to STR(Z6$,1,1). If STR(Z6$,1,1)="N" then POS("YyNn" = STR(Z6$,1,1)) = 3.

PRINT AT(x,y) - Causes the cursor on the CRT to be positioned at (x,y) where x = row number and y = column number.

PRINT HEX() - Special control characters which cause the following operations:

HEX(03) - Clears the CRT screen

HEX(07) - Rings the bell

HEX(0C) - Skip to top of form on printer

HEX(0D) - Carriage return

HEX(0E) - Shift up which causes expanded print on some printers

PRINT TAB() - Causes the printer to skip to the column specified by the value in parenthesis.

PRINTUSING - Prints a line using the variables specified and formating them according to the specifications on the print line.

Example: PRINTUSING 5490,F3,F4

This statement prints the image on line 5490 and inserts F3 and F4 according to the format specified on line 5490.

REM - This statement is used to enter remarks in the program.

SELECT - This statement selects a device and assigns a file number to that device.

Example: SELECT #1 D33

Assigns file #1 to device D33.

SELECT PRINT - This statement assigns the device for print output.

Example: SELECT PRINT 204(132)

Assigns print output to printer 204 with a line width of 132 characters. Device 005 is the CRT.

SELECT P - Causes the system to pause each time a carriage return is sent to the CRT. The number after P specifies the length of pause in increments of 1/6 second.

Example: SELECT P3

Causes a 1/2 second pause. SELECT P0 removes the current pause.

STR() - Specifies an alpha string of characters.

Example: STR(T2$,1,3)

This statement specifies the string of characters in T2$ beginning at position 1 for a length of 3 characters. Trailing spaces are included in the STR function.

$TRAN - This causes certain characters in an alpha-variable to be translated into other characters determined by a table look-up procedure.

Example: If L1$="AaBbCc" and T2$="abc" then after $TRAN(T2$,L1$)R T2$="ABC". L1$ specifies a table of "to-from" pairs with the first character being the value that will be translated to and the second character the value that will be translated from. If a value in T2$ is not in the table then it remains unchanged.

$UNPACK - Unpacks an alpha string into an array.

Example: $UNPACK (D=HEX(0320)) T9$ TO T8$()

This instruction copies T9$ into T8$(). The delimiter D tells where to separate T9$ into elements. The HEX(0320) is a code that says start a new element each time a space is reached and treat consecutive spaces as a single space and ignore leading spaces. Thus, if T9$="A O E" then after $UNPACK, T8$(1)="A", T8$(2)="O", T8$(3)="E".

VAL - This instruction converts a 1 or 2 byte hex value to a Wang numeric value. It can be used wherever a numeric variable can be used.

Example: VAL(B1$(1),2) converts the first two bytes of B1$(1) to a numeric value that can be used in BASIC operations.

< > - Symbol for not equal to.